

AD-A035 563

RFNSSELAER POLYTECHNIC INST TROY N Y COMPUTER RESEAR--ETC F/G 9/2  
COMPUTER RECONSTRUCTION OF BODIES BOUNDED BY QUADRIC SURFACES F--ETC(1)  
SEP 76 R SHAPIRA

UNCLASSIFIED

CRL-48

AFOSR-TR-77-0051

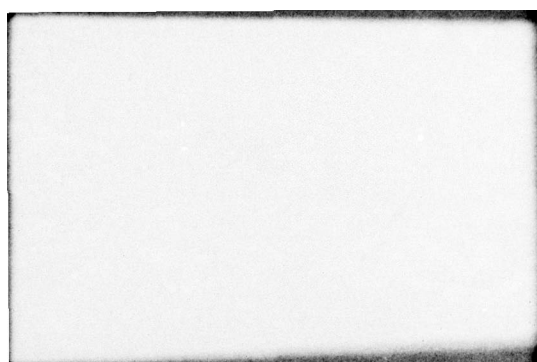
NL

1 of 2  
AD  
A035563









Technical Report CRL-48

Computer Reconstruction of Bodies  
Bounded by Quadric Surfaces from a  
Set of Imperfect Projections

by

Ruth Shapira

September 1976

Approved for public release;  
distribution unlimited.



Prepared for

Directorate of Mathematical and Information Sciences  
Air Force Office of Scientific Research  
Air Force Systems Command, USAF

Grant Number AFOSR 76-2937

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**

Computer Research Laboratory  
Electrical and Systems Engineering Department  
Rensselaer Polytechnic Institute

TROY, NEW YORK 12181

DISTRIBUTION BY	
HTIC	White Section <input checked="" type="checkbox"/>
DTIC	Blue Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

## ABSTRACT

↙ This thesis describes a computer program for constructing a description of solid bodies from a set of  $n$  pictures of the bodies. The bodies are assumed to be bounded by faces which are quadric or planar, and they are restricted to have all their vertices formed by exactly three faces. The pictures are taken from different vantage points, with the restriction that a slight shift in vantage point will not alter the topology of the picture. It is assumed that the program receives outline information from a preprocessor which has extracted this information from the pictures. The outline information (set of line structures) may be imperfect in that some junctions may be erroneously reported and some lines may be missing. However, all lines due to shadows are assumed to have been eliminated by the preprocessor. ↗

The thesis includes a technique for establishing the validity of the junctions presented by the preprocessor as well as for matching corresponding features in the line structures derived from the different pictures. New grammar rules for line-drawing projections of curved and planar solid bodies are developed. These are useful in parsing the line drawings. They have also led to the definition of a new family of impossible objects. The program works simultaneously with all the available line structures. The parsing of every line structure is supported dynamically by the results gotten thus far from the parsing of the other line structures. Through the parsing of the line structures, the use of picture comparison and the application of the grammar rules, many of the preprocessor errors



are detected and partly corrected. The program also can provide feedback to the preprocessor in the form of suggestions as to where to look again for lines in the pictures.

The program utilizes the extracted line structures corresponding to the different bodies in all the pictures to determine the set of faces (insofar as possible) for every body. Every face is defined by an ordered set of  $n$ -tuples. The  $n$ -tuples are the matched lines and junctions in the  $n$  different pictures. The three-dimensional coordinates of the vertices and the equations of the faces can then be determined from these  $n$ -tuples. The program was written in PL/I and has been tested on several scenes.

# ACKNOWLEDGMENT

The author wishes to express her gratitude to Professor Herbert Freeman for suggesting the topic of this research and for his guidance and encouragement.

The author is also grateful to Professors I. Cederbaum and A. Ginzburg of the Technion - Israel Institute of Technology - for their help.

This work was supported by the Air Force Office of Scientific Research, Directorate of Mathematical and Information Sciences, under grants AFOSR 70-1854, AFOSR 75-2755, and AFOSR 76-2937. It was also partly supported by a grant from the American Society for Technion.



## TABLE OF CONTENTS

	<u>Page</u>
Abstract	1
Acknowledgment	iii
List of Illustrations	vi
List of Symbols	viii
I. Introduction	1
1.1 The Problem of Scene Analysis	1
1.2 Previous Work	3
1.3 Basic Definitions and Assumptions	4
a) Definitions relating to the bodies in the scene	4
b) Preprocessor and input data	6
II. Grammatic Properties	8
2.1 Junction Types	8
2.2 Cyclic Order Property	11
III. Geometric Considerations	18
3.1 The Picture Taking Process	18
3.2 Transformation Techniques	21
3.3 Matching Incomplete Data in Three Pictures	22
3.4 Junction Matching	23
IV. Line Matching and Data Recovery	36
4.1 Introduction	36
4.2 Fitting a Line to a Set of Points	37
4.3 Line Matching	38
4.4 Data Recovery	46
4.5 Example	52
4.6 Matching Lines by Range	57
V. Final Description	60
5.1 Region Assembly and Second-Level Data Recovery	60
5.2 Object Formation	65
5.3 Type of Face Determination	66
5.4 Face Equations	67

	<u>Page</u>
VI. Experimental Results	74
6.1 First Example	74
6.2 Second Example	77
6.3 Third Example	77
VII. Summary and Conclusions	100
7.1 Summary	100
7.2 Specific Contributions	102
7.3 Directions for Future Research	103
References	105
Index of Definitions	107



## LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
2.1	Junction Types and Inconsistency of Line Labelling	10
2.2	Illustration of Cyclic Order and Line Assembly	12
2.3	Forced Cyclic Order	15
2.4	Impossible Objects	17
3.1	The Picture-Taking System	19
3.2	The Match Line	25
3.3	The Match-Line Triangle for Vertex J	26
3.4	Triple Ambiguity ( $A_{kq}, B_{iq}, D_{jq}$ )	27
3.5	Ambiguity Problem Arising from Acuteness of Angles between Match Lines	29
3.6	Picture 1 of Illustrative Example	32
3.7	Picture 2 of Illustrative Example	33
3.8	Picture 3 of Illustrative Example	34
4.1	Case-1 Match Configurations (3-Line Junctions)	40
4.2	Matching of Middle Lines in the Configurations Involving Junctions Connected by Three Lines	43
4.3	Case-2 Match Configuration (One 3-Line and One 2-Line Junction)	44
4.4	Case-3 Match Configuration (Two 2-Line Junctions)	45
4.5	Data Recovery Situation	48
4.6	Use of a Synthetic Junction	53
4.7	Schematic Description of the Pictures Shown in Figures 3.6, 3.7 and 3.8	54
4.8	Matching Lines by Range	59



<u>Figure</u>		<u>Page</u>
5.1	Face Group Assembly	61
5.2	Three Configurations for Second-Level Data Recovery	64
5.3	Tree Search for Minimal Cover to Determine Curved-Face Face Group	68
5.4	Curved-Face Determination (Note that Face Group B will Consist of Two Line Assemblies and Two Limbs)	69
5.5	Flow Chart for Determination of Type of Quadric Surface	70
6.1	First Example (The Circled Junctions are Cyclically Arranged. The Arrow Point at Data Recovery Results. Broken Lines Mean "Empty Lines," for which Only the End Points are Known.)	75
6.2	Second Scene Example	78
6.3	Second Scene: Picture-1 Input Data	79
6.4	Second Scene: Picture-2 Input Data	80
6.5	Second Scene: Picture-3 Input Data	81
6.6	Example 2	82
6.7	Computer Output Data for Example 2	83
6.8	Third Scene Example	89
6.9	Third Scene: Picture-1 Input Data	90
6.10	Third Scene: Picture-2 Input Data	91
6.11	Third Scene: Picture-3 Input Data	92
6.12	Example 3	93
6.13	Computer Output Data for Example 3	94

## LIST OF SYMBOLS

$A_{i,s}$	Junction number $s$ in picture $i$
$AB$	Line connecting junction $A$ and $B$
$\underline{a}$	The vector $a$
$C_i$	Center of projection for picture plane $i$
$C_{ijq}$	Intersection between line $C_i C_j$ and plane $P_q$
$F_i$	Distance from $C_i$ to $P_i$
$G_i$	Distance from $C_i$ to center of corresponding film plane frame
$J_{iq}$	Intersection between line $A_{i,J} C_i$ and plane $P_q$
$i=1,2,\dots,n$	Picture plane index (for $n$ pictures)
$i=0$	Index for points not associated with any picture plane
$j=1,2,\dots,n$	Index for coordinate system associated with picture plane $j$
$j=0$	Index for global coordinate system
$\underline{M}_i$	Projection matrix for projecting onto picture plane $i$
$P_i$	Picture plane $i$
$\underline{T}_i$	Matrix of transformation from global system to system of picture plane $i$
$X^j, Y^j, Z^j$	Axes of coordinate system associated with picture plane $i$
$X^j_{i,k}, Y^j_{i,k}, Z^j_{i,k}$	Coordinates of point $k$ associated with picture plane $i$ in coordinate system $j$
$\theta_i$	Angle of rotation of $X^i$ axis from $X^0$ axis
$\phi_i$	Angle of rotation of $Y^i$ axis from $X^0 Y^0$ plane
$\overline{16_1, 12_1}$	Line joining junctions $16$ and $12$ in picture $1$



## I. INTRODUCTION

### 1.1 The Problem of Scene Analysis

When we look at a good quality picture of a scene containing a number of three-dimensional objects, we are usually able to "understand" the scene; that is, we are able to perceive the real physical nature of the objects. The reason for this is that we have seen similar scenes before and at those times were able, by a combination of touch and additional views of the scene, to develop the ability of relating pictures of three-dimensional objects to the objects themselves. The same applies -- perhaps with the need for slightly more specialized learning -- to the understanding of scenes depicted in terms of line drawings, such as a draftsman might prepare. A picture (or line drawing) of a scene is merely a sentence in a language that we have learned. As with other languages, difficulties with understanding arise if the sentence contains words whose meaning is not known to us or if it contains unfamiliar syntactical structures. Further, sudden severe changes in context will disorient us because "picture languages," like other natural languages, are strongly context-dependent.

A scene may be incomprehensible from a picture because the picture contains errors. For example a line drawing (which is a specialized type of picture) may have some line segments missing. In such a case understanding of the scene can sometimes be gained by using a set of pictures (or set of line drawings) describing the scene from different vantage points. Although the pictures may each be defective in some way, collectively they may make possible the unique interpretation of the scene.

In computer scene analysis an image of a three-dimensional scene is converted into an array of pixels, with each assigned a number corresponding to the average grey value in that local area of the image. The image is then preprocessed to extract features suitable for scene interpretation. One common scheme is to extract the lines that are believed to correspond to the edges of the objects. Because of glare, haze, shadows, texture and other lighting conditions, some edges may be partially or wholly obscured and, for basically similar reasons, some spurious lines may be mistakenly thought to correspond to edges. The scene analysis program must extract as much useful information as possible from each picture and then relate this information to similar information devised from other pictures of the same scene (i.e., from pictures taken from different vantage points or under different lighting conditions). The use of multiple pictures is useful both for obtaining information about the three-dimensional nature of the objects as well as for resolving ambiguities due to imperfectly extracted edge data.

We can describe complex bodies in terms of volumes, that is, as assemblies of simple bodies, or we can describe them in terms of their surfaces, using faces, edges and vertices. The second type of description is particularly effective for polyhedra. But it is also suitable for curved bodies, especially man-made bodies, for which every distinct face can be described by a relatively simple mathematical equation. This thesis is restricted to the analysis of scenes containing only bodies with planar or quadric faces; however, the results could be easily modified to deal also with bodies whose curved faces are close to quadric surfaces, and this includes a big subset of man-made bodies.



## 1.2 Previous Work

The description scheme based on the use of volumes is exemplified by Agin (1) who described curved bodies as combinations of generalized cylinders, each described by its axis and cross section. Schemes using the faces-edges-vertices approach are almost entirely limited to polyhedra; they include Roberts (11), Guzman (7), Falk (6), Huffman (8), Clowes (4), Waltz (18), and many more. Nearly all of them utilized only a single picture of the scene. Although all were aware of the pre-processor limitations in extracting complete and valid edge data, the approach varied. Some maintained that a program analyzing an (unrealistic) perfect line drawing also contributes to understanding the process of perception, and indeed, their work provided much insight and guidance for computer scene analysis. They include Guzman (7) with his classification of junctions, Huffman (8) and his elegantly expressed edge labelling scheme, Clowes (4) with his special notation and Waltz (18), who made extensive use of grammatic rules for analyzing perfect line drawings with shadows. There is also the other group, exemplified by Falk (6), who assumed that the input data was imperfectly extracted from real pictures and imposed severe restrictions on the scene (such as preknowledge of the set of bodies). It is this writer's opinion that the restriction to data from a single picture by all these researchers created unnecessary difficulties and caused much effort to be expended on finding solutions to the wrong problems.

The extension of the work from polyhedra to curved bodies, even if only to a restricted family of curved bodies, increases the difficulties to an extent which may not be apparent at first glance. Some of these

difficulties are pointed out in this thesis. A partial extension to curved bodies can be found in the work by Chien and Chang (3). They dealt with bodies whose faces are planar, conic or cylindric, in which every curved face must be bounded by planar faces. They assumed a perfect line drawing for their input data. Their thesis can be considered an extension of Guzman's work.

As already mentioned, virtually all of the early work utilized data from only one picture of the scene. The use of multiple pictures taken from different vantage points was described in the recently published paper by Underwood and Coates (19). Their program analyzes convex polyhedra photographed from different vantage points, a single polyhedron in every set, where the input data is assumed to be perfect. The set of multiple pictures is thus not used to verify unreliable data but merely "to have a look" on all sides of the body.

In this thesis we shall show how a program can be made to "understand" a set of bodies with planar or quadric faces, utilizing defective data extracted from a set of multiple photographs. No preknowledge of the bodies is assumed, but certain general properties of the scene - such as the existence of precisely three faces in every vertex - are assumed.

### 1.3 Basic Definitions and Assumptions

a) Definitions relating to the bodies in the scene. Many of the following terms have been adapted from those used by Woon (20) and Clowes (4).



A quadric surface is the locus of points obeying the equation

$$Q(x,y,z) \equiv \underline{u}^T \underline{A} \underline{u} = 0 \quad (1)$$

where  $\underline{u} = (x, y, z, 1)^T$  is a vector and  $\underline{A}$  is the coefficients matrix

$$\underline{A} = \begin{bmatrix} a_{11} & \frac{1}{2} a_{12} & \frac{1}{2} a_{13} & \frac{1}{2} a_{14} \\ \frac{1}{2} a_{12} & a_{22} & \frac{1}{2} a_{23} & \frac{1}{2} a_{24} \\ \frac{1}{2} a_{13} & \frac{1}{2} a_{23} & a_{33} & \frac{1}{2} a_{34} \\ \frac{1}{2} a_{14} & \frac{1}{2} a_{24} & \frac{1}{2} a_{34} & a_{44} \end{bmatrix} \quad (2)$$

A surface is either a quadric surface or a plane.

An edge is all or part of the intersection of two surfaces; the intersection is bounded by one or two other surfaces or is closed on itself.

A vertex is the intersection of three or more edges.

A face is a portion of a surface bounded by edges or closed on itself. A boundary is a closed chain of edges bounding a face. A single face may have several boundaries.

A body is a closed, connected part of the 3D space, delimited by a finite number of faces.

A scene is a set of bodies.

A projection is a central projection of a 3D entity onto the picture plane.

A limb is the locus of the points on a quadric surface that are tangent to the projecting rays and do not lie on an edge.

A virtual vertex is the intersection of an edge and a limb.

A region is a connected, visible part of the projection of a face. The projection of a face may result in zero, one, or more regions.

A line is the projection of an edge or a limb. The line projected by a limb will also be called a limb when this will not cause any confusion. A line may be straight or curved.

A junction is the intersection of two or more lines, and is thus either the projection of a vertex or of a virtual vertex, or is a result of obscuring part of an edge or limb by a face.

An object is the set of regions, lines, and junctions corresponding to a single body.

We impose the following restrictions:

1. Every vertex in the scene is formed by exactly three surfaces, and belongs to exactly three edges.
2. Smooth transition between two different faces is not allowed (i.e., the derivative must be discontinuous across the edge).
3. The camera position is assumed to be "general." For example, the projections of different vertices may not coincide in any picture. If a vertex does not belong to an edge, the projection of the vertex may not coincide with the projection of the edge in any picture.
4. No limb passes through a vertex.

b) Preprocessor and input data. It is assumed that there is a preprocessor which extracts the lines and junctions from the picture (photograph) of the scene. The extraction of curved lines has been the subject of work by Ramer (10) and Agin (1). There is also the work of



Shirai and Tsuji (15), who extracted straight lines from photographs of polyhedra, getting rid of shadows at the expense of missing some visible lines. Our preprocessor is assumed to be a combination of Shirai and Tsuji's (ORing and ANDing of a sequence of pictures taken from the same vantage point under different conditions of illumination to eliminate shadows) and of Ramer's (curved-line extraction and determination of junctions). The output of such a preprocessor is the input to the scene description program described here. The assumptions about the input data thus are as follows:

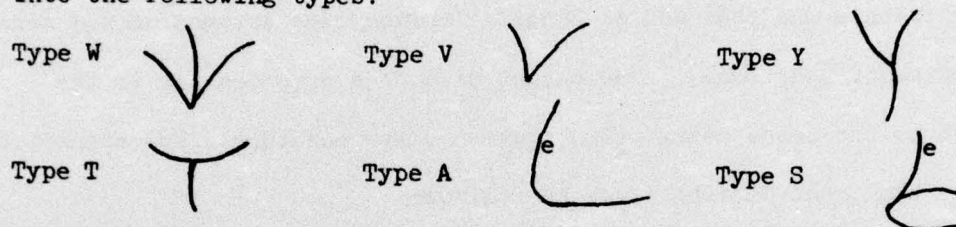
1. There are no extraneous lines.
2. Some valid lines (or parts of them) may be missing.

Since the kind of preprocessor assumed here was not in fact available, the data for testing of the scene description program had to be extracted visually from sets of actual photographs.




## II. GRAMMATIC PROPERTIES

### 2.1 Junction Types

In accordance with the assumptions stated in the previous section, junctions can have no more than three lines. They can thus be classified into the following types:



Types W, V, Y and T are the well-known types, generalized here (7).

Types Y and W must be projections of vertices. Type T results from the covering of part of an edge or limb by a face. Types A and S are the projections of virtual vertices, where the line e is the projection of a limb. Characteristic to junction types A and S is that all the lines in the junction have a common tangent. A type-V junction is either the projection of a vertex whose third line is not visible or it is a junction which could be of either type Y or W but has one line missing, or of type T with part of the bar missing. If an S junction should be missing a line, it can look like a type-A junction , like a type-V junction , or be undetectable . (The dots indicate the missing line.) An illustration of the different junction types is given in Fig. 2.1.

The detection of a type-A junction is difficult since this type of junction is characterized by a continuous transition between two lines. The two lines are governed by different equations and on the basis of



this segmentation of the continuous contour into the two line segments should be possible. Such segmentation was accomplished by Agin (1) and Albano (2).

Through junctions of types S and A, a limb can be identified. A limb's importance lies in that it is projection-dependent and has no match in a picture taken from a different vantage point. Also, it conveys information regarding the type of the face (cylinder, cone, ellipsoid, hyperboloid, etc.) and is useful in finding the face's equation. The type identification of each junction is assumed to be supplied by the preprocessor.

It would be useful to have an extended (or generalized) junction labelling scheme based on Huffman's polyhedral labelling ("- for a line projected by a concave edge, "+" for a convex edge, and a directed arrow for a convex edge created by a front and a back face\*). However, for curved objects the labelling of a line may not always be consistent and, therefore, the usefulness of junction labelling here is of doubtful value since it cannot be propagated along the line. (See for example the labelled line in Fig. 2.1). The distinction in this thesis between a Y and a W junction is in keeping with past practice rather than for any practical reason.

---

\* A front face is a face for which the outward normal (at least at the edge in question) has a component directed toward the vantage point. For a back face, the component is directed away from the vantage point. See Woon and Freeman (21).

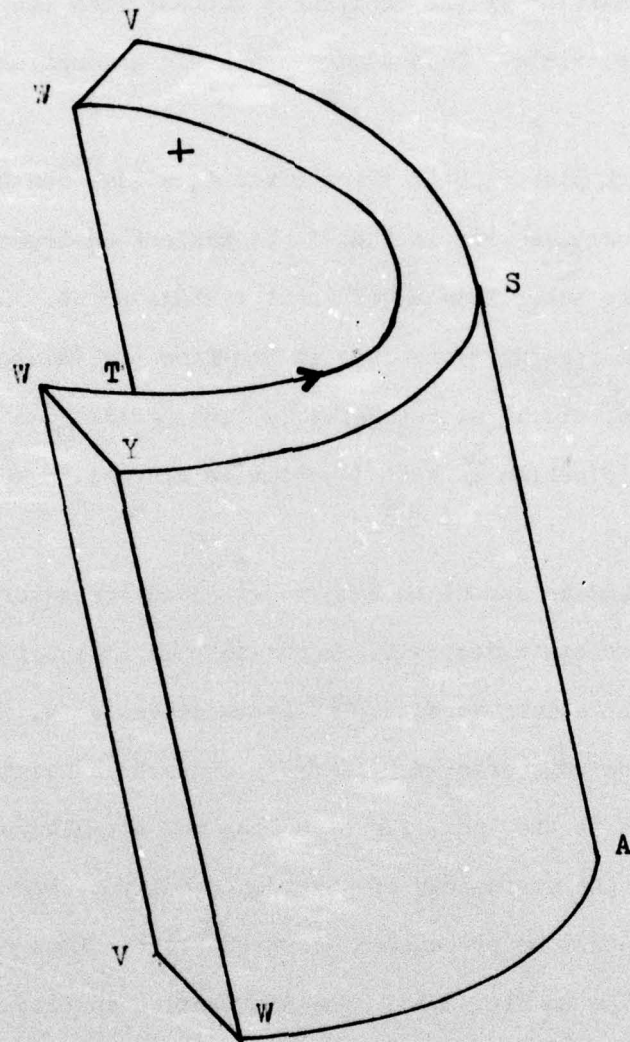


Fig. 2.1 Junction Types and Inconsistency of Line Labelling



## 2.2 Cyclic-Order Property

Let us define the depth index of a line as the number of faces modulo-2 between the edge corresponding to this line and the center of projection. As mentioned before, every vertex belongs to exactly three edges. We define for these edges a cyclic order in the vertex, as the order induced by "walking around" the vertex in a clockwise manner, and numbering the edges in the order 1, 2, 3. The cyclic order will retain its clockwise sense in any projection, irrespective of the shape of the vertex, under either of the following two conditions:

1. All three lines have an even depth index.
2. One of the three lines has an even depth index and two have an odd one.

The cyclic order will be reversed (become counter-clockwise) in a projection if:

1. One of the lines has an odd depth index and two have an even one.

or

2. All three lines have an odd depth index.

The different cases are illustrated by the example shown in Fig. 2.2.

If we trace out the edges of a boundary, we consistently change edges at the vertices, either always in a decreasing cyclic order or always in an increasing cyclic order. Let us always choose to trace edges in an increasing order. Then we shall traverse every edge in two different directions as we walk around the two faces that share it.

(See the two arrowed lines along the line AB in Fig. 2.2).

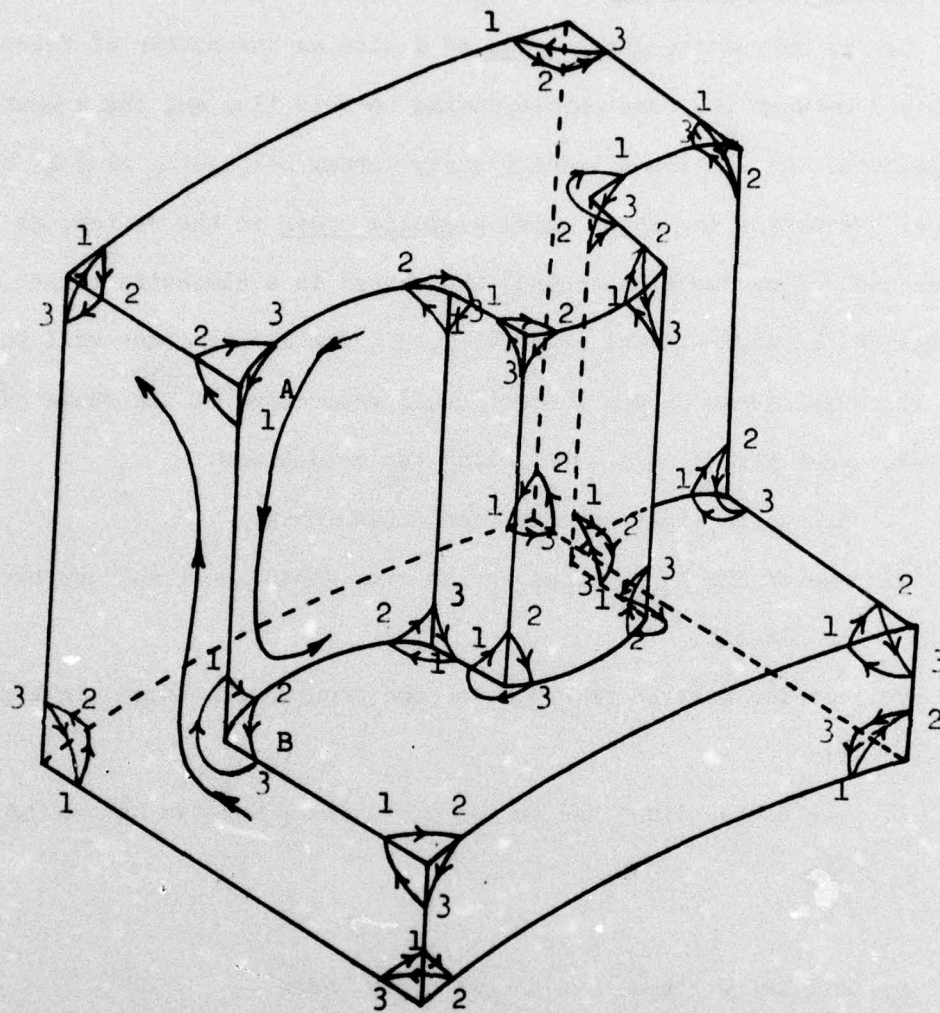


Fig. 2.2 Illustration of Cyclic Order and Line Assembly



From the foregoing we conclude that when a vertex's projection has three lines, their cyclic order is determined by their arrangement in the picture. However, when one line is missing, the cyclic order is not known. Let us define the relation  $AB < AC$  to mean that in junction A the line AC follows the line AB immediately in the cyclic order. Then if A is a two-line junction, we have either the relation  $AB < AC$  or the relation  $AC < AB$ .

Knowledge of the cyclic order at a junction is crucial to the assembly of the lines belonging to a single region. Therefore, it is important to have a means for determining the cyclic order in a junction when the order is not given naturally. Some rules and techniques are stated in this section, with additional rules given later in the thesis.

We define a line assembly (LA) as the directed path followed in tracing out the lines corresponding to a single boundary - or any continuous part of it - in increasing cyclic order. We denote an LA in picture i by the ordered set of junctions  $A_{i,1} \dots A_{i,n}$  visited in the course of the trace. Now it follows from the definition that for every  $1 < k < n$  we have  $A_{i,k} A_{i,(k-1)} < A_{i,k} A_{i,(k+1)}$ . Two LA's are said to be distinct if they trace out lines corresponding to two distinct boundaries. If for two LA's,  $A_{i,1} \dots A_{i,n}$  and  $A_{j,1} \dots A_{j,m}$

1.  $A_{i,1} = A_{i,n}$ , and there is at least one  $A_j$  different from every  $A_i$

or

2. There are two successive junctions in the two LA's such that

$$A_{i,k} = A_{j,\ell} \text{ and } A_{i,(k-1)} = A_{j,(\ell+1)}$$

or

3.  $A_{i,n} = A_{j,1}$  and the relation  $A_{j,1} A_{j,2} < A_{i,(n-1)} A_{i,n}$  exists.

Then the two LA's are distinct.

Two rules forcing a cyclic order in a junction can be stated now:

Rule 1: If we have an LA,  $A_{i,1} \dots A_{i,n}$ , in which  $A_{i,1} = A_{i,n}$  then we must have in  $A_{i,1}$  the relation  $A_{i,1} A_{i,(n-1)} < A_{i,1} A_{i,2}$ .

For example, in Fig. 2.3(a) we have the LA B,C,D,A,B, (The LA is shown as an arrowed line). The lines in B are thus forced to have the relation

$$BA < BC$$

Rule 2: If we have two distinct LA's,  $A_{i,1} \dots A_{i,n}$  and

$A_{j,1} \dots A_{j,m}$ , such that  $A_{j,m} = A_{i,1}$  and  $A_{j,(m-1)} \neq A_{i,2}$ , then we must have in  $A_{i,1}$  the relation  $A_{i,1} A_{i,2} < A_{j,m} A_{j,(m-1)}$ .

For example in Fig. 2.3(b) we have the two distinct LA's: E,F,C,B and B,A,G,E. They are distinct because of the cyclic order already established in junction B by Rule 1. Now by Rule 2 the relation  $EF < EG$  must hold in E. (This in turn establishes the LA M,F,E,G,N,O).

A more complicated strategy may be adopted to force the correct order at a junction when these two rules cannot be applied directly. One assumes a cyclic order in the junction and tries to reach a contradiction by a sequence of inductive steps. If a contradiction is reached, the cyclic order opposite to the one selected is forced on the junction. For example, in Fig. 2.3(c) we assume an order in junction R such that  $RQ < RN$ . Then we can establish the LA's P,Q,R,N,G,A,D and D,C,F,M,P. They are distinct as implied by the natural order in D, and hence by Rule 2 force in P the relation  $PQ < PM$ . At the same time we have the two LA's R,Q,O and O,N,R



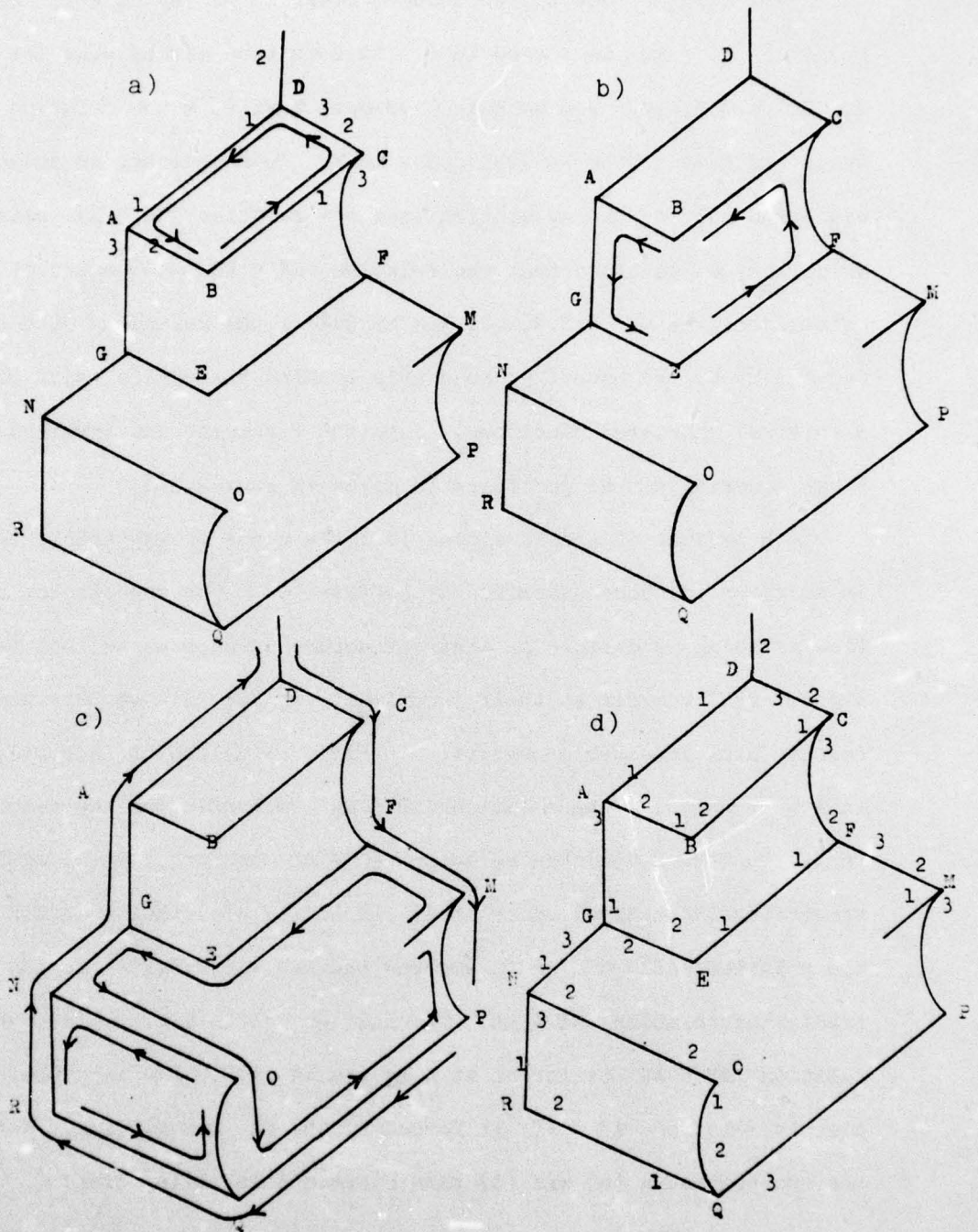


Fig. 2.3 Forced Cyclic Order

which are distinct due to the assumed order in R. By Rule 2, the relation  $ON < OQ$  is forced in O. This in turn establishes the LA  $M, F, E, G, N, O, Q, P, M$ . Now by Rule 1 we must have in M the relation  $MP < MF$ , which contradicts the natural order in M. This sequence of deductions was based only on the assumption that the relation  $RQ < RN$  exists at R, which lets us conclude that the relation  $RN < RQ$  must exist at R. This establishes the LA  $O, N, R, Q, O$ , and by Rule 1 the relation  $OQ < ON$  is induced at O. We have thus been able to find the cyclic order at 4 of the 5 original unordered junctions. Junction P remains undetermined. The final ordering of the junctions is given in Fig. 2.3(d).

A byproduct of the foregoing is a new class of impossible objects, in addition to those described by Huffman (8). The objects can be identified as being impossible if their structure is such as to lead to contradictory cyclic order at their junctions. In Fig. 2.4 we have some instances of impossible objects. In Fig. 2.4(a) the LA  $A, E, D, C, B, A$  forces in A - by Rule 1 - the relation  $AB < AE$  contradicting the natural order in A. In Fig. 2.4(b) the relation  $AD < AB$  is forced in A, again contradicting the natural order at A. In Fig. 2.4(c) the LA  $A, E, F, B, A$  forces the relation  $AB < AE$  in A, whereas the LA  $A, B, C, D, E, A$  forces the contradicting relation  $AE < AB$ . The same occurs in Fig. 2.4(d), where the relation  $AB < AE$  is forced at A by the LA  $A, E, F, B, A$  and the contradictory relation  $AE < AB$  is forced by the LA  $A, B, C, D, E, A$ . Note that the two polyhedra (a) and (d) pass Huffman's labelling test.



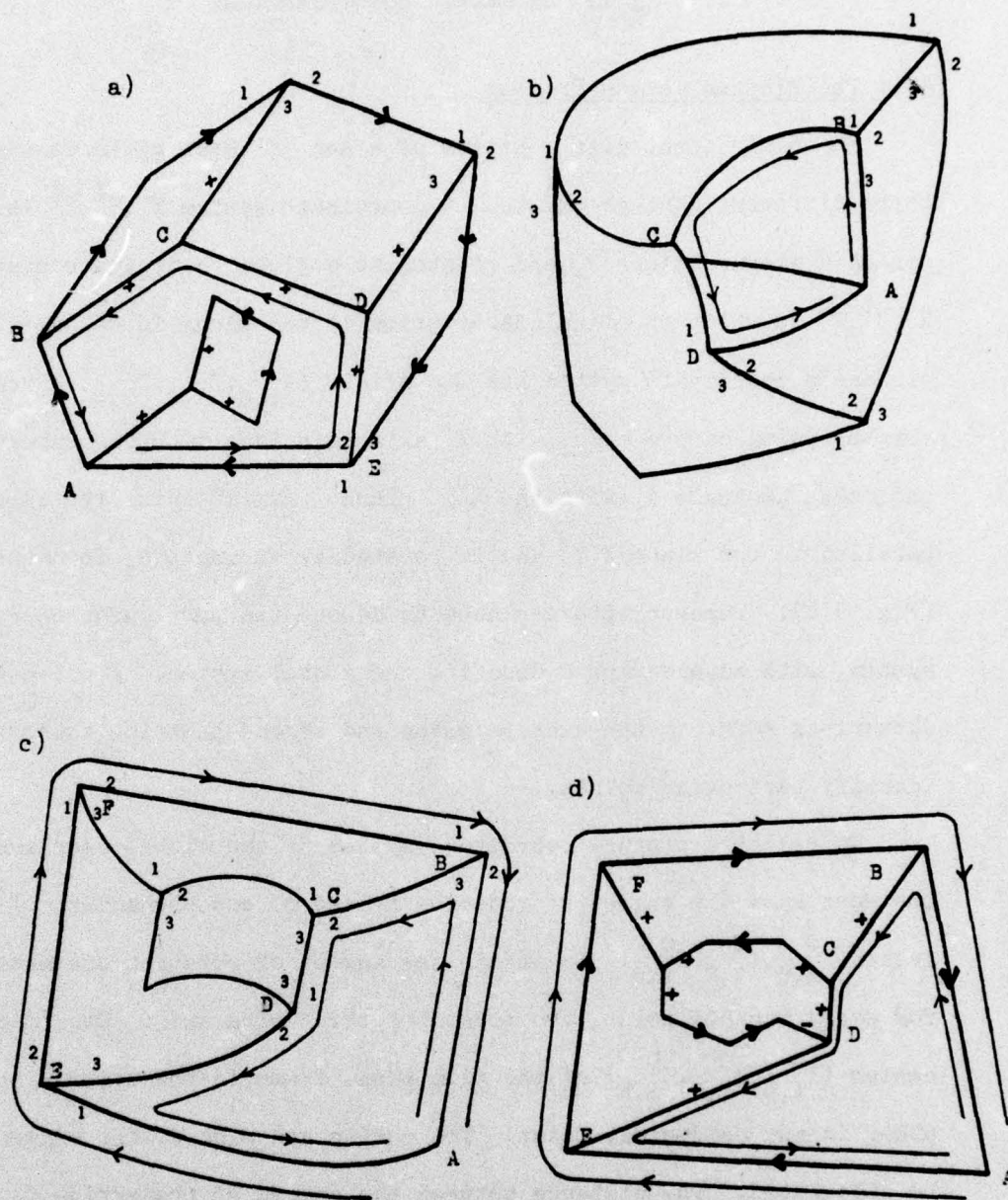


Fig. 2.4 Impossible Objects

### III. GEOMETRIC CONSIDERATIONS

#### 3.1 The Picture Taking Process

The basic input data consists of a set of three pictures taken from three different vantage points. A coordinate system  $X^i, Y^i, Z^i$  is defined for each picture plane  $P_i$  and related to a global coordinate system  $X^0, Y^0, Z^0$  in which the final description of the scene is determined. Each picture's coordinate system has its origin  $(X_{i,0}^0, Y_{i,0}^0, Z_{i,0}^0)$  on the corresponding camera axis. Its  $Y^i$  axis coincides with the camera axis and makes an angle  $\phi_i$  with the  $X^0 Y^0$  plane. Its  $X^i$  axis lies in a plane parallel to the plane  $X^0 Y^0$  and is rotated by an angle  $\theta_i$  from the  $X^0$  axis (Fig. 3.1). Superscripts are used to denote the particular coordinate system, with superscript 0 denoting the global system. First-position subscripts refer to the picture plane and second-position subscripts identify particular points.

To relate a picture coordinate system to the global coordinate system one must know the angles of rotation  $\theta_i$  and  $\phi_i$  and the amount of translation  $X_{i,0}^0, Y_{i,0}^0, Z_{i,0}^0$ . To obtain the angles of rotation one measures, for every vantage point, two points on the camera axis. One point is the center  $(X_{i,N}^0, Y_{i,N}^0, Z_{i,N}^0)$  of the film plane frame in the camera, and the other is any convenient point. The cosine and sine of the angles can now be calculated. The distance between the center of projection  $C_i$  and the point  $(X_{i,N}^0, Y_{i,N}^0, Z_{i,N}^0)$  is denoted by  $G_i$ . The distance between the center of projection and the point  $(X_{i,0}^0, Y_{i,0}^0, Z_{i,0}^0)$  is denoted by  $F_i$ . The ratio  $R_i = G_i/F_i$  is determined by measuring corresponding diagonals of both



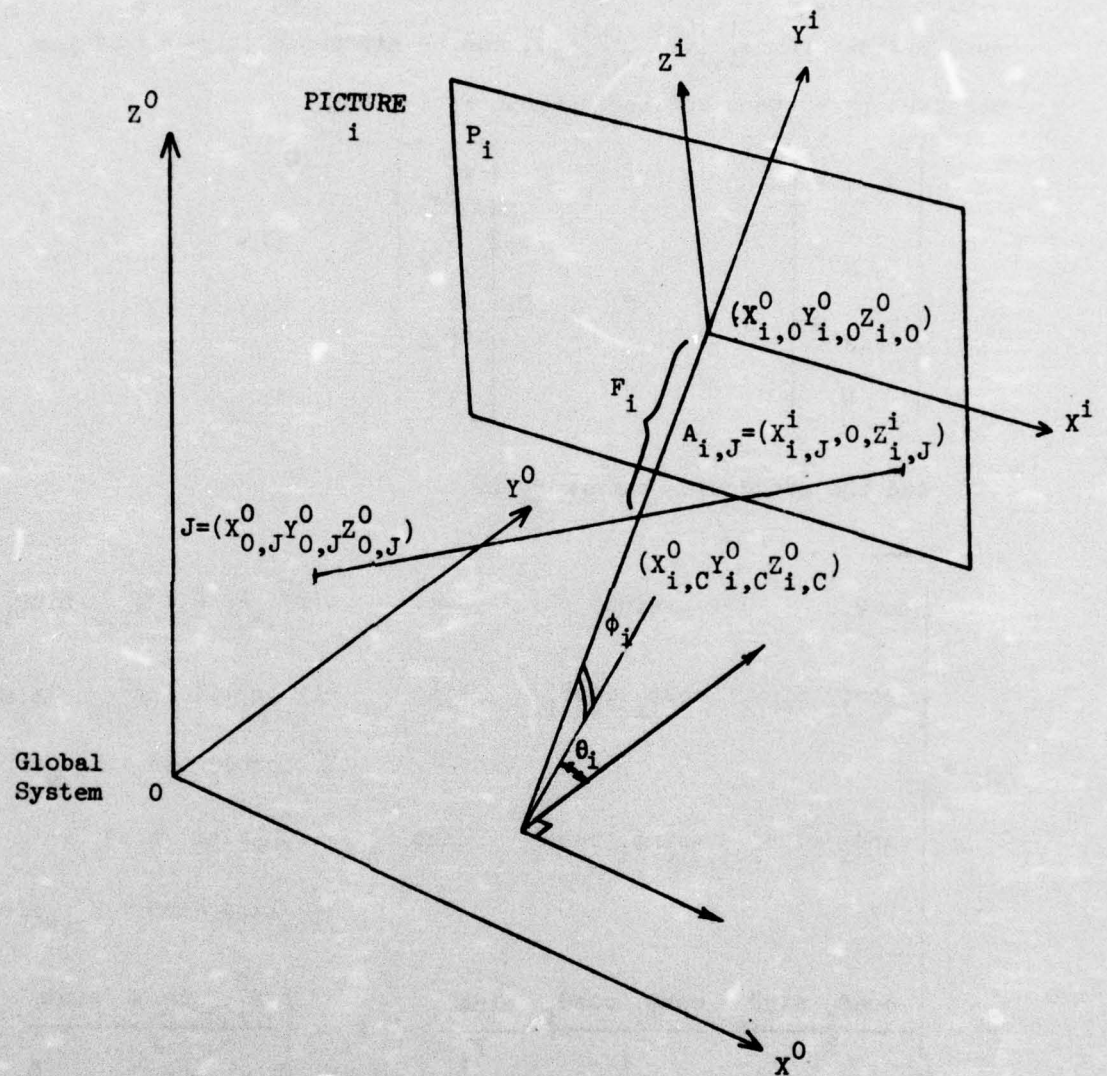


Fig. 3.1 The Picture-Taking System

the film plane frame and the uniformly enlarged picture. The relation between a point J in space, whose coordinates in the global system are  $(X_{0,J}^0, Y_{0,J}^0, Z_{0,J}^0)$ , and whose projection coordinates in a picture coordinate system are  $(X_{i,J}^i, Y_{i,J}^i, Z_{i,J}^i)$ , can be expressed in terms of the measured parameters and the unknown  $F_i$  (5,14):

$$\begin{bmatrix} X_{i,J}^i & W \\ Y_{i,J}^i & W \\ Z_{i,J}^i & W \\ & W \end{bmatrix} = \underline{M}_i \begin{bmatrix} X_{0,J}^0 \\ Y_{0,J}^0 \\ Z_{0,J}^0 \\ 1 \end{bmatrix} \quad (3)$$

And the projection matrix  $\underline{M}_i$  is

$$\underline{M}_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & -(X_{i,N}^0 \cos\theta_i + Y_{i,N}^0 \sin\theta_i) \\ -\cos\phi_i \sin\theta_i & \cos\phi_i \cos\theta_i & \sin\phi_i & -(F_i(R_i+1) + X_{i,N}^0 \cos\phi_i \sin\theta_i + Y_{i,N}^0 \cos\phi_i \cos\theta_i + Z_{i,N}^0 \sin\phi_i) \\ \sin\phi_i \sin\theta_i & -\sin\phi_i \cos\theta_i & \cos\phi_i & -(X_{i,N}^0 \sin\phi_i \sin\theta_i - Y_{i,N}^0 \sin\phi_i \cos\theta_i + Z_{i,N}^0 \cos\phi_i) \\ \frac{-\cos\phi_i \sin\theta_i}{F_i} & \frac{\cos\phi_i \cos\theta_i}{F_i} & \frac{\sin\phi_i}{F_i} & -R_i - \left( \frac{-X_{i,N}^0 \cos\phi_i \sin\theta_i}{F_i} + \frac{Y_{i,N}^0 \cos\phi_i \cos\theta_i + Z_{i,N}^0 \sin\phi_i}{F_i} \right) \end{bmatrix} \quad (4)$$



In order to compute  $F_i$  for every picture we look at a set of pre-measured points  $\{(X_{0,j}^0, Y_{0,j}^0, Z_{0,j}^0)\}$ ,  $j=1 \dots n$ , whose set of projections  $\{(\hat{X}_{i,j}^i, \hat{Z}_{i,j}^i)\}$   $j=1 \dots n$ ,  $i=1 \dots 3$ , are measured in every picture. To minimize the sum-of-squares differences between the calculated projections and the measured projections we require that

$$\frac{\partial}{\partial F_i} \left( \sum_{j=1}^n (X_{i,j}^i - \hat{X}_{i,j}^i)^2 + \sum_{j=1}^n (Z_{i,j}^i - \hat{Z}_{i,j}^i)^2 \right) = 0, \quad i=1 \dots 3 \quad (5)$$

for every picture. The three equations can now be solved for  $F_i$ .

### 3.2 Transformation Techniques

Interrelation between the three picture coordinate systems can be achieved by transforming the coordinates via the global system. The translation  $(X_{i,0}^0, Y_{i,0}^0, Z_{i,0}^0)$  can be expressed as follows:

$$\begin{bmatrix} X_{i,0}^0 \\ Y_{i,0}^0 \\ Z_{i,0}^0 \end{bmatrix} = \begin{bmatrix} X_{i,N}^0 \\ Y_{i,N}^0 \\ Z_{i,N}^0 \end{bmatrix} + F_i (R_i + 1) \begin{bmatrix} -\cos\phi_i \sin\theta_i \\ \cos\phi_i \cos\theta_i \\ \sin\phi_i \end{bmatrix} \quad (6)$$

The matrix of transformation from the global system to the system of pictures  $i$  is:

$$T_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & -(X_{i,0}^0 \cos\theta_i + Y_{i,0}^0 \sin\theta_i) \\ -\cos\phi_i \sin\theta_i & \cos\phi_i \cos\theta_i & \sin\phi_i & -(-X_{i,0}^0 \cos\phi_i \sin\theta_i + Y_{i,0}^0 \cos\phi_i \cos\theta_i + Z_{i,0}^0 \sin\phi_i) \\ -\sin\phi_i \cos\theta_i & -\sin\phi_i \cos\theta_i & \cos\phi_i & -(X_{i,0}^0 \sin\phi_i \sin\theta_i - Y_{i,0}^0 \sin\phi_i \cos\theta_i + Z_{i,0}^0 \cos\phi_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Thus if  $x_{0,j}^j, y_{0,j}^j, z_{0,j}^j$  are the coordinates of a point expressed in the coordinate system of picture  $j$  and  $x_{0,j}^i, y_{0,j}^i, z_{0,j}^i$  are the coordinates of the same point in the coordinate system of picture  $i$ , we have

$$\begin{bmatrix} x_{0,j}^i \\ y_{0,j}^i \\ z_{0,j}^i \\ 1 \end{bmatrix} = \underline{T}_i \underline{T}_j^{-1} \begin{bmatrix} x_{0,j}^j \\ y_{0,j}^j \\ z_{0,j}^j \\ 1 \end{bmatrix}$$

Other parameters of importance are the coordinates  $(x_{i,C}^0, y_{i,C}^0, z_{i,C}^0)$  of the center of projection (center of lens)  $C_i$  for every picture:

$$\begin{bmatrix} x_{i,C}^0 \\ y_{i,C}^0 \\ z_{i,C}^0 \end{bmatrix} = \begin{bmatrix} x_{i,N}^0 \\ y_{i,N}^0 \\ z_{i,N}^0 \end{bmatrix} + F_i R_i \begin{bmatrix} -\cos\phi_i \sin\theta_i \\ \cos\phi_i \cos\theta_i \\ \sin\phi_i \end{bmatrix} \quad (8)$$

### 3.3 Matching Incomplete Data in Three Pictures

The line structure data extracted from the pictures may be incomplete because of occlusion or because of preprocessor failure. Let us consider the data imperfections due to preprocessor failure. They can be divided into two groups:

(1) Data imperfections involving junctions.

a. Missing junctions.

b. False junctions - for example, when half of the bar of a

T junction is missing, causing it to look like a V junction.



- c. A V junction that is actually a Y or W junction with one line missing.

Note that the V in subgroup c, although of wrong type, is the projection of a vertex; this is not true for V in b.

(2) Data imperfections involving lines.

- a. Part of a line is missing and the broken end is open (dangling).
- b. Part of a line is missing and a junction of wrong type is created at the broken end.
- c. The whole line is missing.

We call a junction valid when it is a projection of a vertex.

Thus, junctions of types A, S, and T are not valid, junctions of type Y and W are valid and for V junctions we do not know. The precise nature of a V junction must be left undecided until more evidence is collected, as will be explained later.

A junction of type S which is missing a line and is reported as an A junction, does not cause any confusion and, therefore, is not counted as an imperfection.

### 3.4 Junction Matching

In order to arrive at a true description of the 3D scene in spite of some data imperfections, we compare the line structures from the three pictures against each other and use the information found in one picture to check the information found in another.

Let us start with junctions. We have two goals. One is to determine which of the V junctions are valid, or, to put it differently for which V junction is there enough evidence to support such an assumption. The

second is to group, from the different pictures, junctions that are the projection of the same vertex. From here on in this section whenever we use the term "junction" we shall mean a V, Y or W junction.

If two junctions, one each in two different pictures, are projections of the same vertex, they must obey certain geometric rules. Let  $P_i$ ,  $C_i$  and  $A_{i,J}$  be the picture plane, the center of projection, and the projection of vertex J, respectively, for picture i. Then for two pictures, i and j,  $C_i$ ,  $C_j$ , J,  $A_{i,J}$ , and  $A_{j,J}$  are coplanar, and the lines  $C_i A_{i,J}$ ,  $C_j A_{j,J}$  intersect any plane  $P_q$  in three colinear points:  $C_{iq}$ ,  $J_{iq}$  and  $J_{jq}$ , respectively (Fig. 3.2). We shall call the line formed by these points a match line. Thus if we re-project into the plane  $P_q$  junction  $A_{i,J}$  from  $C_i$ , junction  $A_{j,J}$  from  $C_j$ , and  $C_i$  from  $C_j$ , we can check whether this condition is satisfied. Two junctions from different pictures that obey this condition are said to be matchable. Obviously, if a vertex I is in the plane  $C_i$ ,  $C_j$ , J then  $A_{i,I}$  and  $A_{j,J}$  are matchable and so are  $A_{i,J}$  and  $A_{j,I}$ . To resolve such ambiguities, we refer to the third picture, picture k. Now we require that  $C_i$ ,  $C_j$ ,  $C_k$  be not colinear. Then if  $A_{i,J}$ ,  $A_{j,J}$ , and  $A_{k,J}$  are the three projections of vertex J, the three different match lines on plane  $P_q$  form a triangle whose vertices are  $J_{iq}$ ,  $J_{jq}$ , and  $J_{kq}$ . (Fig. 3.3). We call the set of three junctions, from different pictures, that are matchable in pairs, a triple. By forming triples we can eliminate many pairs that are matchable but are not projections of the same vertex. However, even in the pure geometric case we may have three projections of three different vertices A, B, and D that form a triple. (Fig. 3.4).



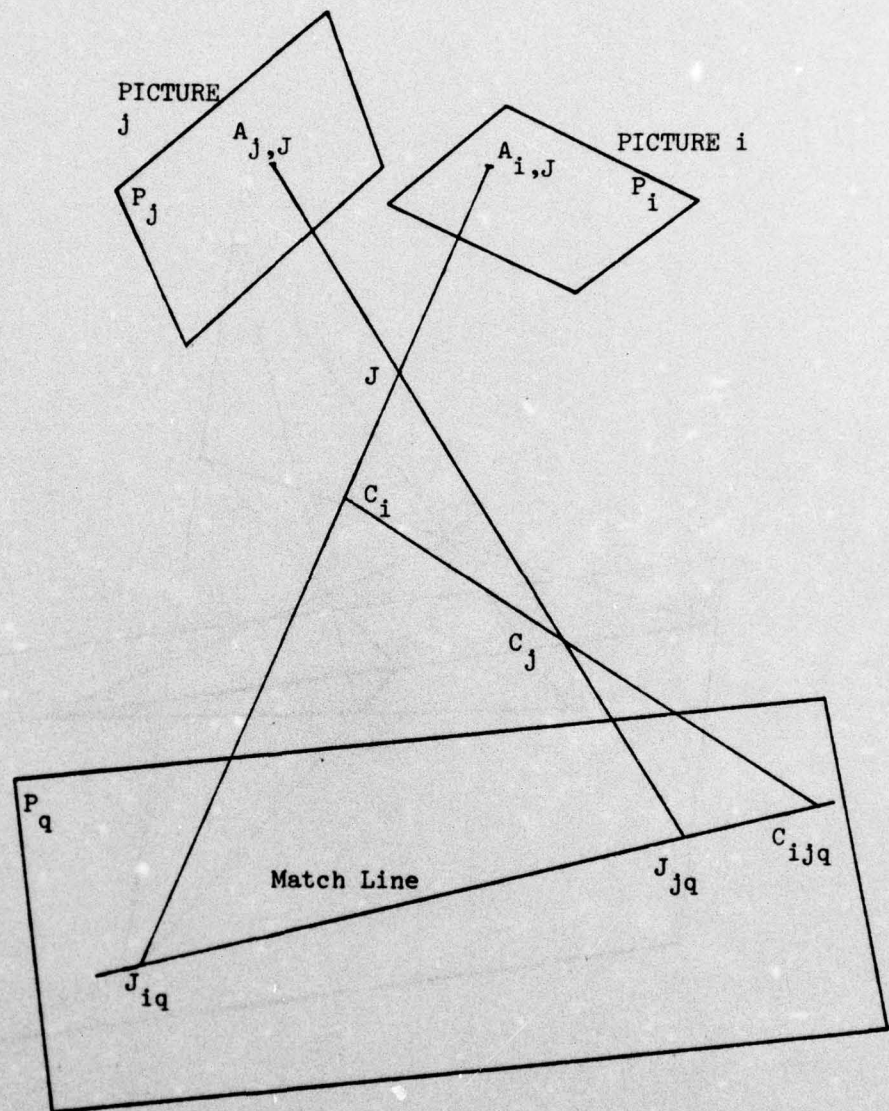


Fig. 3.2 The Match Line



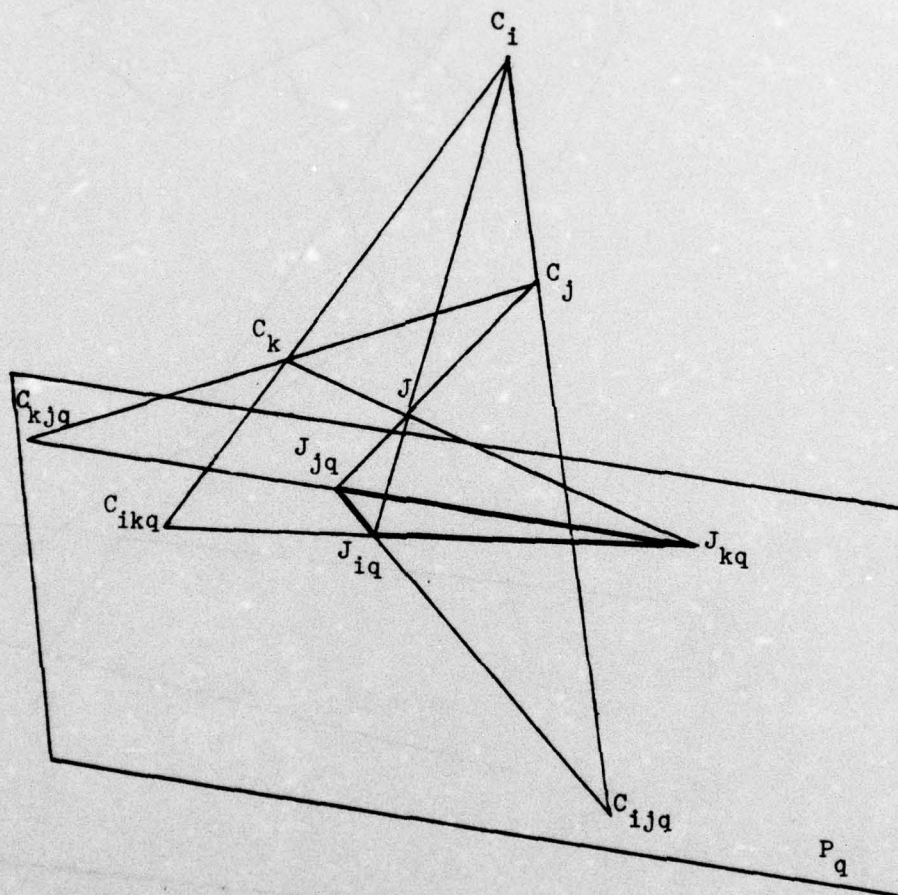


Fig. 3.3 The Match-Line Triangle for Vertex J  
(Match Lines Shown Bold)

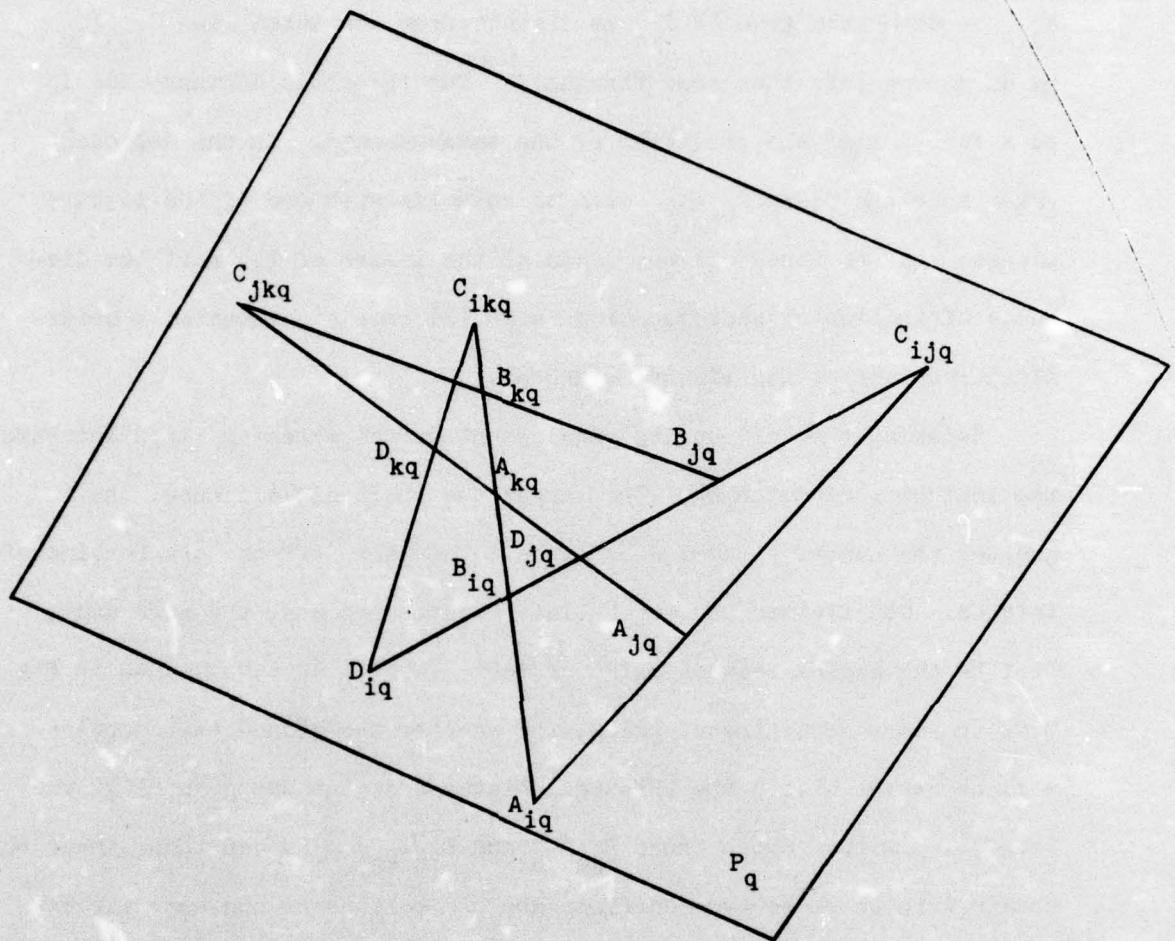


Fig. 3.4 Triple Ambiguity ( $A_{kq}, B_{iq}, D_{jq}$ )



We are dealing here with real data; thus we must accept  $A_{i,J}$  and  $A_{j,J}$  as matchable even if  $J_{iq}$  is distant from the match line  $C_{ijq}^{J_{jq}}$  by an amount less than some threshold. The threshold distance should be a function of the precision of the measurements. In the approach taken here the plane  $P_q$  was taken to coincide with one of the picture planes, and the threshold was taken as the lesser of (1) half the distance of the two closest junctions and (2) some experimentally determined fraction of the picture diagonal.

Relaxing the collinearity requirement on the matching lines increases the ambiguity of matching. The larger the threshold distance, the greater the number of ambiguous cases. This also affects the forming of triples. The flatter the match-line triangle becomes, the more acute will be the angles between pairs of match lines. As can be seen in Fig. 3.5, the more acute these angles, the greater the chance that a point will be caught within the threshold distance of two wrong match lines (e.g.,  $I_{iq}$  within match lines  $B_{jq}^{B_{iq}}$  and  $B_{iq}^{B_{kq}}$ ). We can thus expect to obtain triples where two junctions are projections of the same vertex but the third is not.

Two matchable junctions that are projections of the same vertex are said to match each other. A triple in which the junctions match each other is called a match triple.

To establish matches, we find for every junction all matchable junctions in the two other pictures. Then we form all the triples. To find the match triples in the set of all triples, we have to use some picture context. The more severe the conditions causing ambiguity, the more the need for context information.

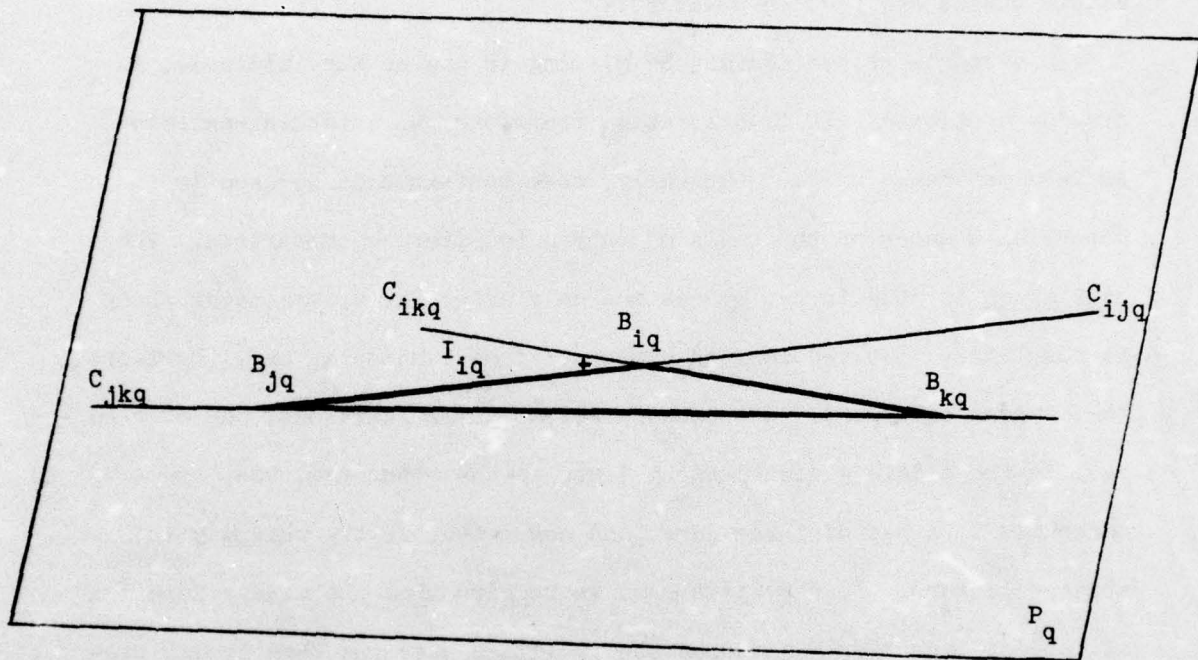


Fig. 3.5 Ambiguity Problem Arising from Acuteness of Angles Between Match Lines. (The wrong triple is created.)



The kind of context we shall use is line connections between junctions. If we have three lines - each in a different picture - whose ends form two triples, the two triples are each considered to be a match triple. (The chance of naming a wrong triple as a match triple still exists but is now reduced greatly).

A vertex's projection may be missing in one or more pictures, as already mentioned. It is desirable, therefore, to establish matches between matchable pairs. Naturally, more context must be used to establish a match on the basis of only a two-picture comparison. The pair match is done in two passes and only after the triple match phase is completed. (It reduces the number of free candidates and, therefore, the chances of error). We select a set of three junctions, one of them - call it the middle - connected by lines to the other two, that are matchable to a set of three junctions connected, in the same way in another picture. In the first pass we require that the middle junction will be of type Y or W and that the two lines have the same cyclic precedence in the two junctions. The three junctions are then assumed to match the three junctions in the other picture. In the second pass we allow one or both of the middle junctions to be of type V and do not check the cyclic precedence (it is not known yet for the two-line junction). The second pass is a little more hazardous, not having the ability to use cyclic precedence, but since it is carried out after most junctions are already matched, there is only a small chance of making an error. We also accept two junctions forming a loop, which are matchable to two other junctions forming a loop, as matching to the other two.

If one junction in a triple is matched to the other two by two applications of a pair-wise match, we consider the triple to be a match triple; that is, we consider the remaining two as matching also.

Every V junction that is matched to a junction in another picture is marked as valid (although it may miss a visible line). The match supplies additional evidence that the junction is a projection of a vertex.

An illustrative example is shown in Fig's. 3.6 to 3.8. The figures consist of three pictures of the same scene, taken from different vantage points. In every picture some lines, or parts of lines are missing. Many Y and W junctions are thus reported as V junctions, but the most misleading information is in the false shape identification of junctions 15 in picture 1, 27 in picture 2 (both are T junctions missing a line and reported as type V) and 24 in picture 2 (an S junction reported as type V).

The program reports many matchable relations. To mention a few: junction 9 in picture 1 is reported matchable to junctions 1,8,10,12,27 in picture 2 and to junction 21 in picture 3. Junction 27 in picture 2 (a false junction) is reported as matchable to junctions 5,9,10 in picture 1, and junction 24 in picture 3 is reported as matchable to junctions 5,6,16 in picture 1 and to junctions 1,7,11,17 in picture 2. Thirteen triples are formed (the subscript in the junction indicates the picture number)



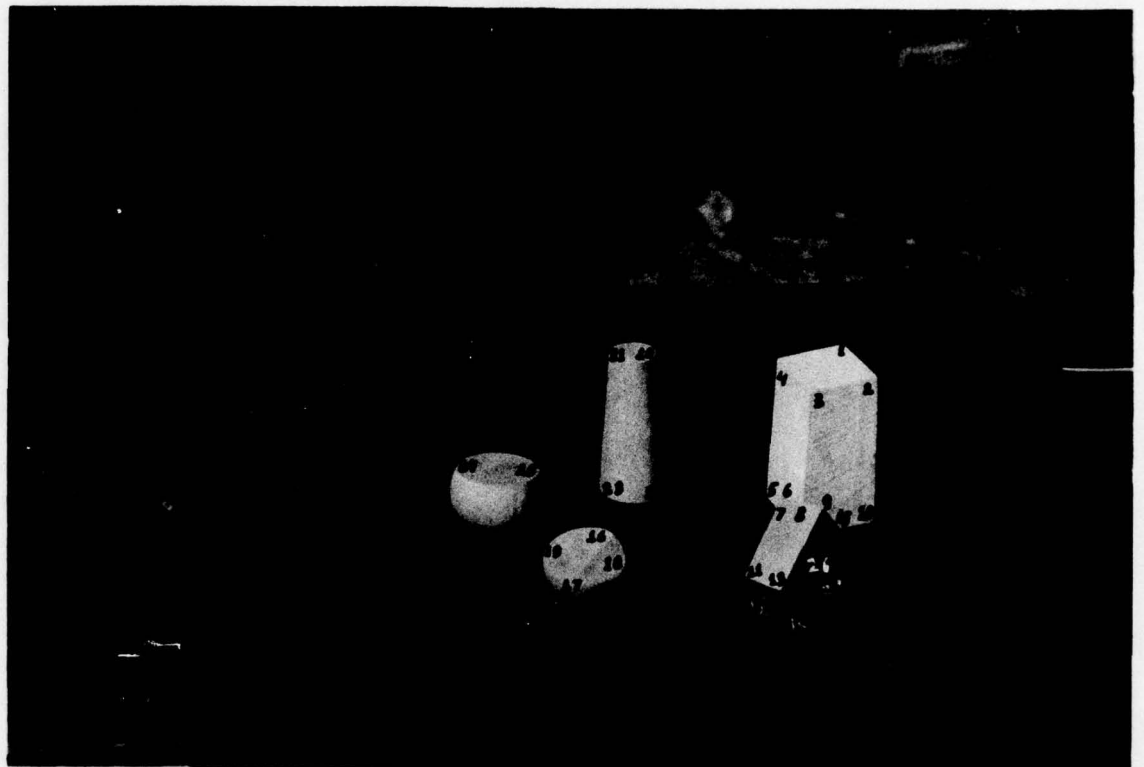


Fig. 3.6 Picture 1 of Illustrative Example

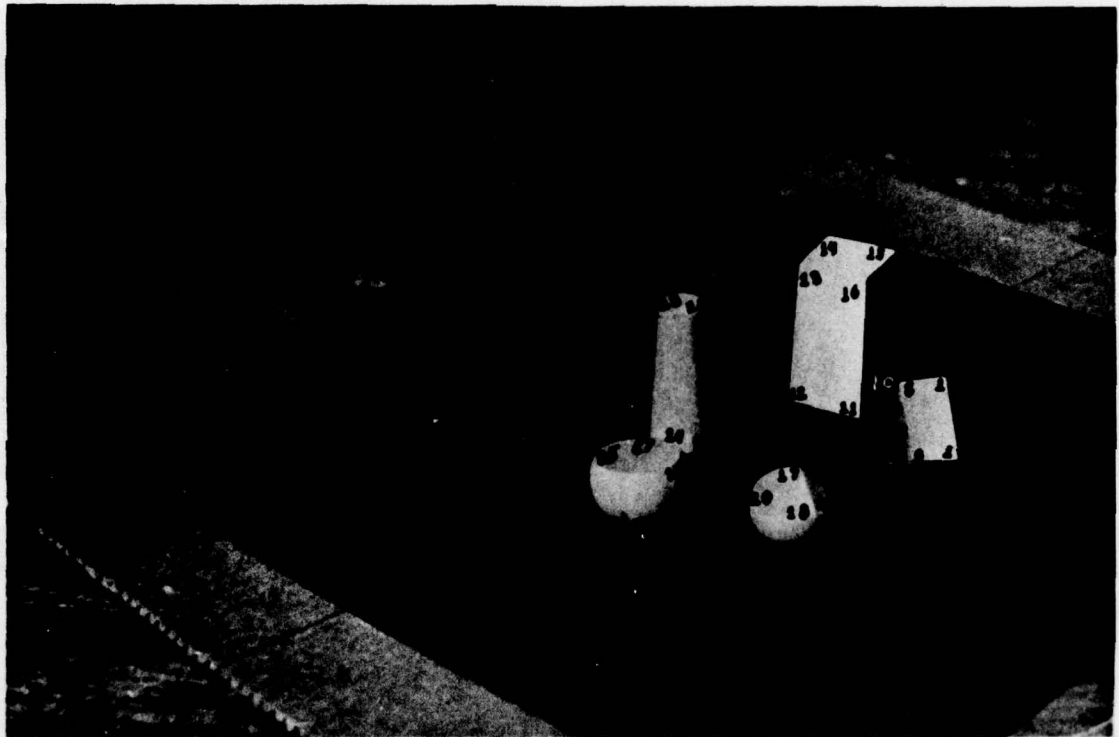


Fig. 3.7 Picture 2 of Illustrative Example



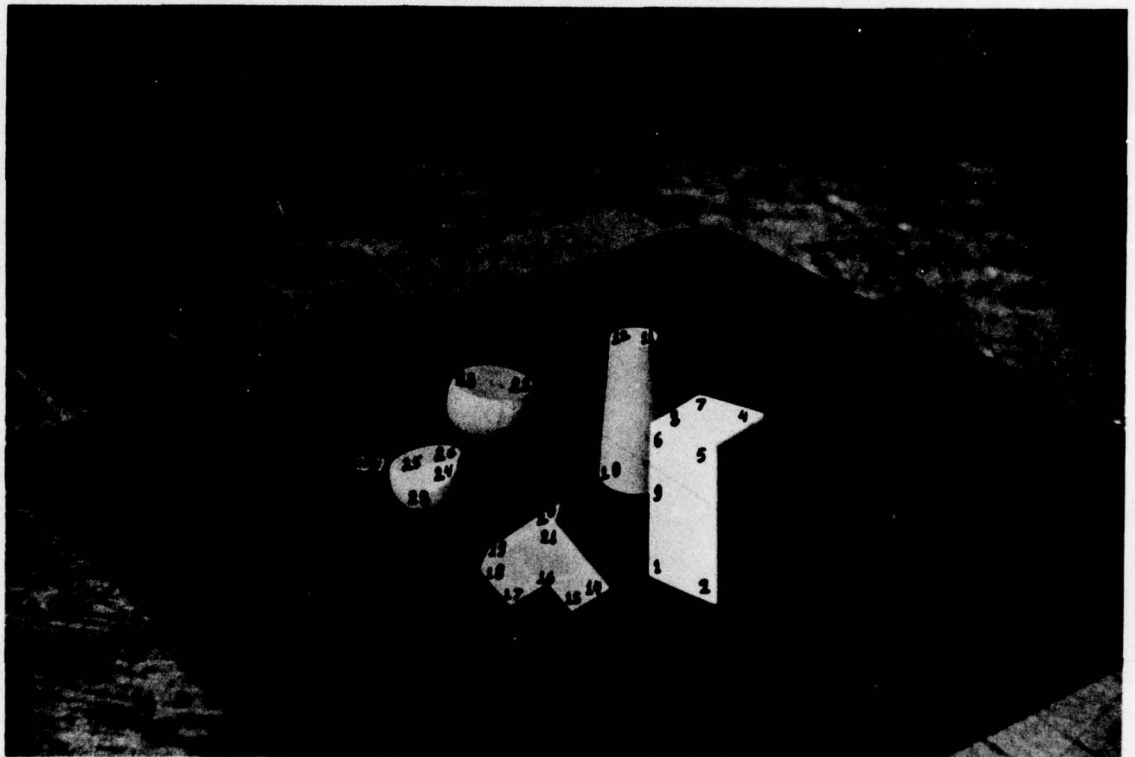


Fig. 3.8 Picture 3 of Illustrative Example

$(5_1, 1_2, 2_3)$	$(9_1, 1_2, 21_3)$	$(14_1, 3_2, 17_3)$
$(3_1, 16_2, 6_3)$	$(6_1, 8_2, 20_3)$	$(4_1, 13_2, 7_3)$
$(6_1, 1_2, 2_3)$	$(11_1, 9_2, 19_3)$	$(1_1, 14_2, 4_3)$
$(16_1, 17_2, 2_3)$	$(13_1, 2_2, 18_3)$	$(2_1, 15_2, 5_3)$
		$(10_1, 10_2, 2_3)$

On the basis of picture context the last nine triples can be accepted as being match triples. Of the four rejected triples, two are the three projections of two vertices, namely  $(3_1, 16_2, 6_3)$  and  $(16_1, 17_2, 2_3)$ , but at this stage no context support is found. This is partly corrected in the next stage where we compare two pictures and establish matches between pairs of junctions. In the first pass we get the matches of the pairs  $(16_1, 17_2)$ ,  $(17_1, 18_2)$ ,  $(12_1, 4_2)$ ,  $(3_1, 16_2)$  and  $(7_2, 13_3)$ . In the second pass we get  $(5_1, 12_2)$ ,  $(11_2, 1_3)$ ,  $(16_3, 26_1)$  and  $(16_2, 6_3)$ . The last one gives, together with the pair  $(3_1, 16_2)$  reported in the first pass, the match triple  $(3_1, 16_2, 6_3)$ . Hence the following V junctions are marked as valid:

$1_1, 3_1, 4_1, 5_1, 6_1, 10_1, 12_1, 17_1, 26_1$

$1_2, 3_2, 7_2, 10_2, 12_2, 13_2, 14_2, 16_2, 18_2$

$1_3, 5_3, 6_3, 7_3, 13_3, 16_3, 17_3, 18_3, 19_3, 21_3$

No support was gained for the wrongly reported junctions

$15_1, 24_2$ , and  $27_2$

nor for the (correctly reported) junctions

$5_2, 6_2, 3_3$  and  $15_3$ .



#### IV. LINE MATCHING AND DATA RECOVERY

##### 4.1 Introduction

Line matching is not a straightforward procedure. The fact that between two junctions we may have one, two, or three lines, complicates the situation, and it becomes even worse in the case of imperfect data. We have to be very careful in matching lines that all the possibilities are taken into consideration.

Line matching also serves as an important tool for recovering data. Data recovery consists basically of three different actions: (1) detection of a missing connection between two junctions, (2) extension of an existing line from one valid junction to another valid junction, and (3) detection of the location of a missing valid junction, validating the one found there (if any), or creating a synthetic junction.

Data recovery, if successful, leads to conditions for further line matching, which in turn enables more data recovery, and so on.

Cyclic order in an unordered valid junction is fixed if conditions for Rule 1 or Rule 2 exist (See Chapter II). Some previously unordered junctions can be ordered on the basis of line matching, as we shall see later. This in turn may create conditions for further application of Rule 1 or Rule 2, and so on.

Line matching, data recovery, and cyclic ordering is done in several passes. Each pass creates new conditions for the next pass until finally a condition is reached in which no new results can be obtained.

#### 4.2 Fitting a Line to a Set of Points

It is important for the procedure to be able to determine the type of line represented by a set of points and sometimes to determine its exact equation as well.

Given a set of points  $(x_i, z_i)$ , we try to fit a line to it so that the sum of the squared errors will be minimized. First we try to fit a straight line

$$a_2 x + a_1 z + a_0 = 0$$

where

$$a_0 = \sum x_i z_i - \frac{\sum x_i^2 \sum z_i^2}{n}$$

$$a_1 = \frac{\sum x_i^2 \sum z_i - \sum x_i z_i \sum x_i}{n}$$

$$a_2 = \frac{\sum x_i \sum z_i^2 - \sum z_i \sum x_i z_i}{n}$$

The average distance of the points from the straight line is calculated and, if it is less than some threshold  $T_1$ , we accept the fit. The threshold is selected on the basis of visual judgment.

If the average distance is greater than some  $T_2 > T_1$ , then we declare the set as representing a curved line. For an average distance lying between these two values, we leave the type undetermined. If the set was declared to represent a curved line and we wish to fit a conic  $g$  to it,

$$g(x, z) \equiv a_5 x^2 + a_4 z^2 + a_3 xz + a_2 x + a_1 z + a_0 = 0$$



then by minimizing the sum of squared errors

$$\sum g^2(x_i, z_i) \equiv \underline{a}^T \cdot \underline{X} \cdot \underline{a}$$

and constraining  $a_0$  to equal 1, we can solve for the rest of the coefficients using  $\underline{X} \cdot \underline{a} = 0$ . Here  $\underline{a}$  is the vector of coefficients and

$$\underline{X} = \begin{bmatrix} \Sigma x_i^4 & \Sigma x_i^2 z_i^2 & \Sigma x_i^3 z_i & \Sigma x_i^3 & \Sigma x_i^2 z_i & \Sigma x_i^2 \\ \Sigma x_i^2 z_i^2 & \Sigma z_i^4 & \Sigma x_i z_i^3 & \Sigma x_i z_i^2 & \Sigma z_i^3 & \Sigma z_i^2 \\ \Sigma x_i^3 z_i & \Sigma x_i z_i^3 & \Sigma x_i^2 z_i^2 & \Sigma x_i^2 z_i & \Sigma x_i z_i^2 & \Sigma x_i z_i \\ \Sigma x_i^3 & \Sigma x_i z_i^2 & \Sigma x_i^2 z_i & \Sigma x_i^2 & \Sigma x_i z_i & \Sigma x_i \\ \Sigma x_i^2 z_i & \Sigma z_i^3 & \Sigma x_i z_i^2 & \Sigma x_i z_i & \Sigma z_i^2 & \Sigma z_i \\ \Sigma x_i^2 & \Sigma z_i^2 & \Sigma x_i z_i & \Sigma x_i & \Sigma z_i & \Sigma 1 \end{bmatrix} \quad (9)$$

The foregoing methods have their drawbacks (11,16,1) but they are good enough for the purpose here.

#### 4.3 Line Matching

The matching of lines is done around matched junctions, taking two pictures at a time. Two lines in two pictures, matching the same line in a third picture, are declared to match each other. With every junction there are associated three indices 1,2, and 3 for the three possible lines that share the junction. (If the junction is cyclically ordered, these indices coincide with the cyclic order.) We say that the junction

is linked in an index t if there is a line corresponding to that index whose other end junction is a valid junction.

The two matched junctions around which we try to match lines fall into the following three cases:

- (1) both contain three lines,
- (2) one contains three lines and the other contains two lines, or
- (3) both contain two lines.

In each of these three cases when one line in one junction is matched to a line in the other junction and both junctions are ordered cyclically, further line matching can be done by proceeding cyclically around the junctions.

At the beginning of the line matching process all the 3L (3 lines) junctions are cyclically ordered according to their natural order, and, therefore, in Case 1, matching one line forces the match of the other two lines. At a later step in the procedure a line may be added to an unordered 2L junction, yielding an unordered 3L junction; then matching one line in a Case-1 situation will not automatically lead to the match of the rest.

Fig. 4.1 shows under what conditions we can match a line in Case 1. The basic idea is to find out whether between a pair of junctions in the same picture there is a single line, or, if more than one, to determine their relations to each other. For example in configuration (a) of Fig. 4.1 it is clear that line  $c_1$  must be the only line between junctions  $I_1$  and  $J_1$  because  $J_1$  is linked in its other two indices. It follows that the line  $d_j$  is the only line between junctions  $I_j$  and  $J_j$ .



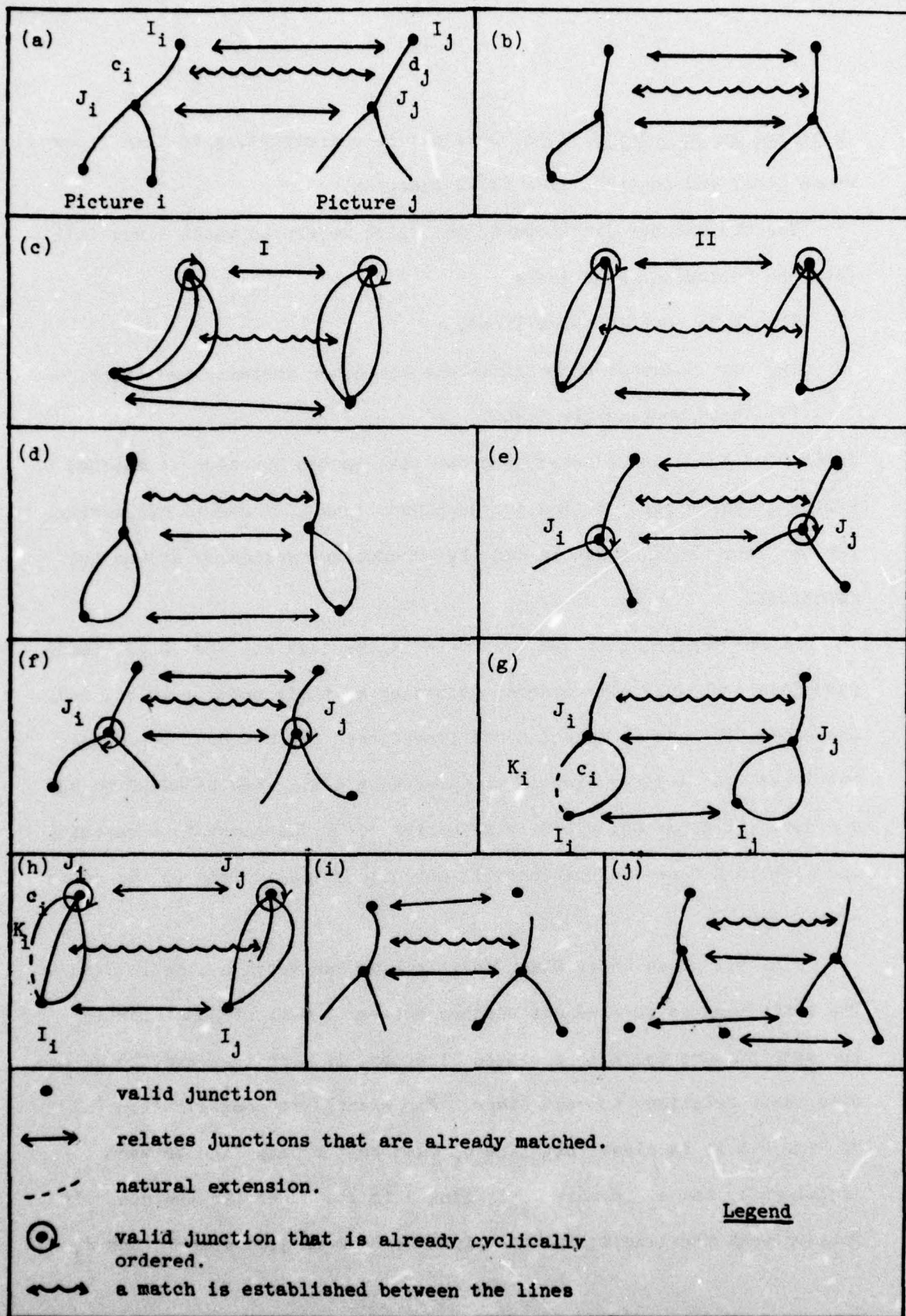


Fig. 4.1 Case-1 Match Configurations (3-Line Junctions)

considering the pairwise matching of the junctions (though it might not be clear from looking at picture  $j$  alone when  $J_j$  is not linked in its two other indices). Thus the result is a match between lines  $c_i$  and  $d_j$ . Part of the requirements in configurations (c), (e), (f) and (h) is that the junctions  $J_i$  and  $J_j$  be cyclically ordered to establish even the first match. Configurations (g), and (h) require some special condition, namely that  $I_i$  lie on the natural extension of the line  $c_i$ . This means that we must successfully fit a line to  $I_i$  and the points of  $c_i$  and that this line comply with the conditions stated in the following paragraph.

As explained in section 4.2 a straight line fit is acceptable if the average miss distance is less than some threshold. If a straight line fit is rejected and a conic is fitted instead, we get either an ellipse, a hyperbola or no locus at all. Other types of conics are not realistic since they involve a vanishing parameter, which in practice does not occur in the computation because of truncation errors. No locus at all means of course that a conic fit has been rejected. We also reject any fit accepted thus far if, (1) the distance between  $I_i$  and  $K_i$  is greater than the distance between  $J_i$  and  $I_i$  ( $K_i$  is the non-valid end junction of  $c_i$ ), (2) if it is an ellipse and one of the axes is too small (to avoid a fit "jumping" from one side of the ellipse to the other), (3) if it is an ellipse and the part of the ellipse between  $K_i$  and  $I_i$  is greater than a quarter of the ellipse, and (4) if it is a hyperbola and the points of the set belong to the two branches of the hyperbola. This tightening of acceptance conditions



very much reduces the possibility of a wrong extension, but at the same time it also rules out some good extensions, especially under condition (3).

Configurations c and h present a special problem since they both deal with junctions connected by three lines. To obtain matches for them we assume that the middle line stays in the middle. Although true in most cases, it is easy to show counterexamples. But even then it is not crucial since the configuration is closed on itself and the error does not propagate. To find the middle line we note that when we walk from one junction to the other and back, changing lines in increasing cyclic order, we trace three closed paths. In two of them the path is traced in a counterclockwise manner (Fig. 4.2), and in one of them in a clockwise manner. The last one happens to be the path where the middle line is not involved. Using this fact the middle lines in both pictures can be detected and matched to each other. The rest follow cyclically.

Fig. 4.3 shows the configurations required for matching lines in Case 2. Note configurations (c) and (j) where a natural extension is a part of the conditions. Note also configurations (d), (k), (e), and (i), which imply that junctions  $J_i$  and  $I_i$  in each should be connected. We shall return to this subject in the next section.

Fig. 4.4 shows the conditions under which we match lines in case 3. Note in both configurations the implication of a missing connection, between  $J_i$  and  $I_i$  and between  $J_j$  and  $K_j$  in (a), and between  $J_j$  and  $I_j$  in (b). In Figs. 4.1 and 4.3 the shape of the three-lines junctions, (whether Y or W) is irrelevant to the configurations.

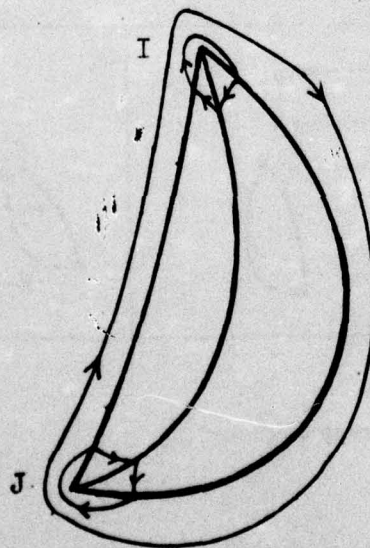
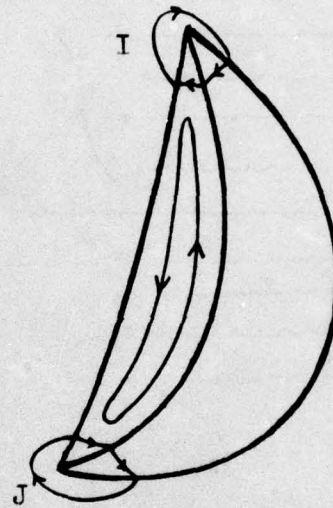
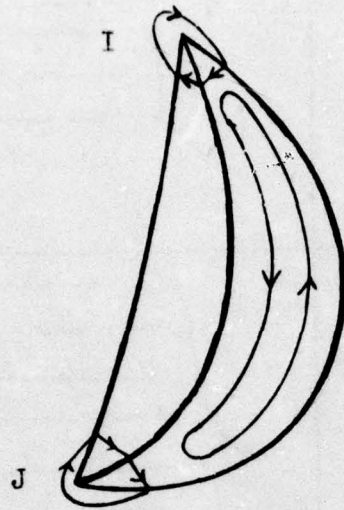


Fig. 4.2 Matching of Middle Lines in Configurations  
Involving Junctions Connected by Three Lines



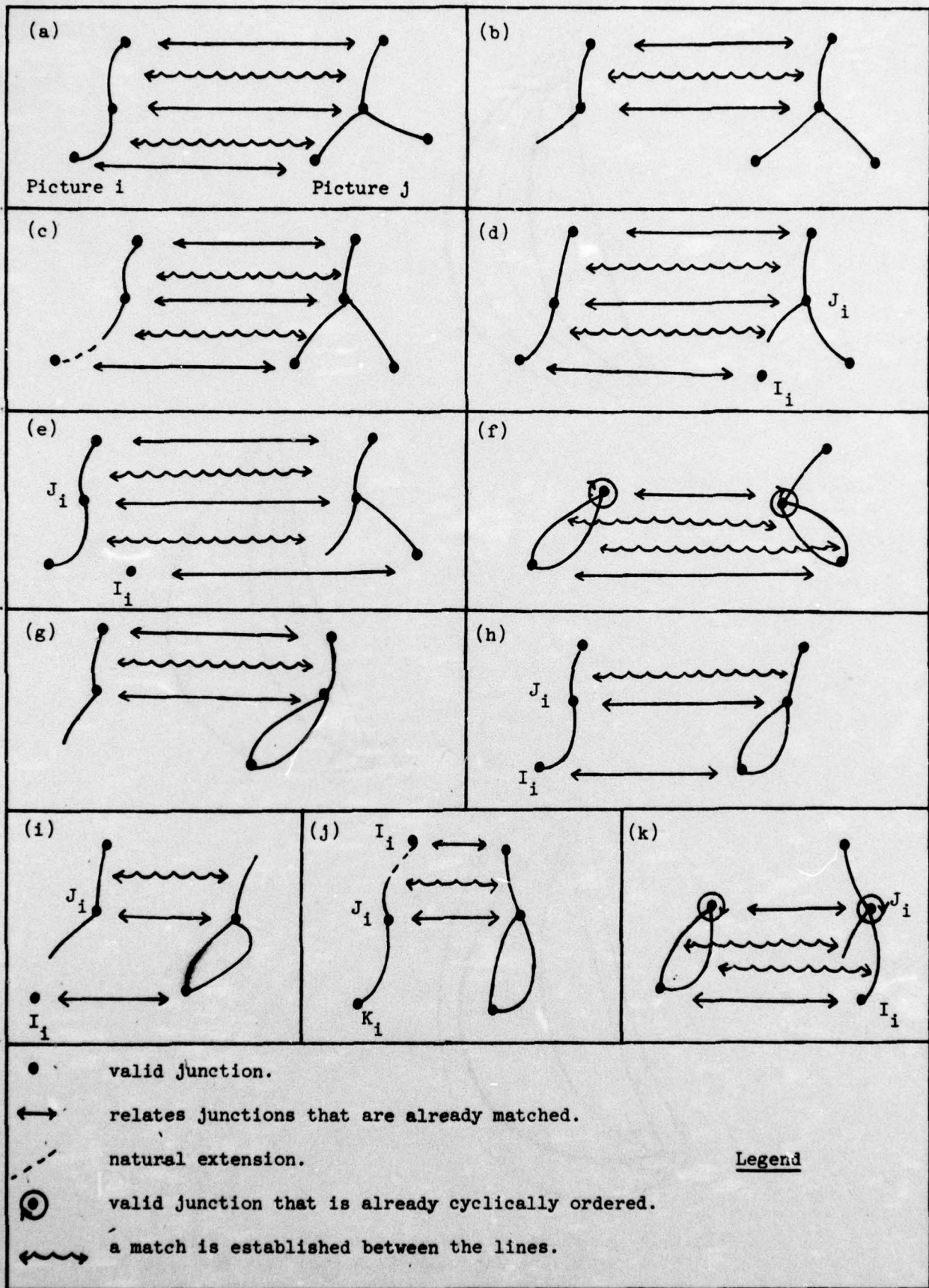


Fig. 4.3 Case-2 Match Configuration (One 3-Line and One 2-Line Junction)

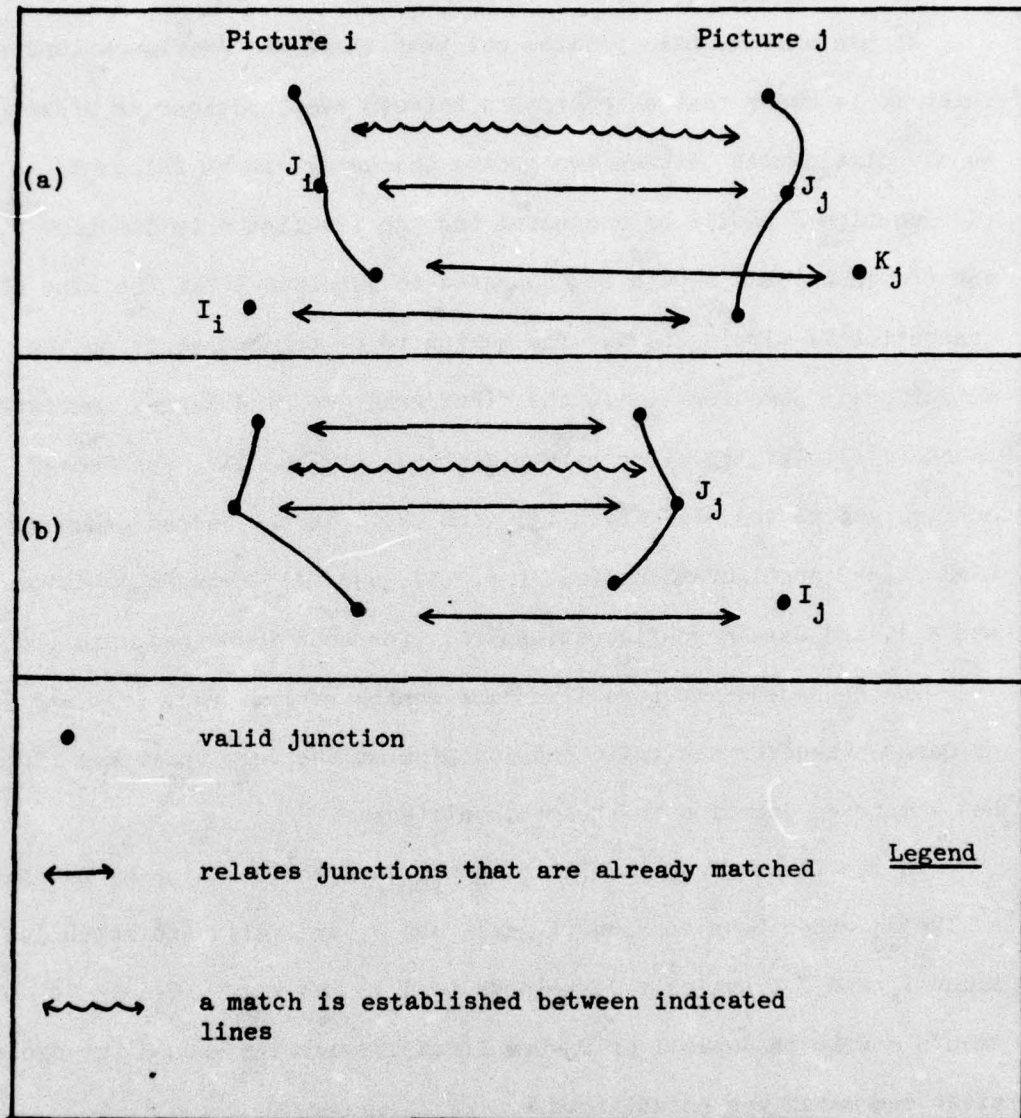


Fig. 4.4 Case-3 Match Configuration (Two 2-Line Junctions)



#### 4.4 Data Recovery

It has already been pointed out that there are configurations for which it is clear that a connection between two junctions is missing. We can distinguish between two groups characterized as follows:

(1) junction J should be connected through its line c to junction I, and (2) junction J should be connected to junction I but the line of connection is missing in J. The action to be taken depends on the situation in junction I. In the first group we find Case-1 configurations (c)II, (g), (h) (for both pictures), (i) and (j), and Case-2 configurations (c), (d), (i), (j), and (k). In the second group we find Case-2 configurations (e), (h), (i), and (j) (between junctions  $J_1$  and  $K_1$ ), and Case-3 configurations (a) (for both pictures), and (b).

In the second group we find also configurations (a), (c), and (d), of Case-2 whenever the valid end junction of the only unmatched line has a matched junction in the other picture.

In the first group we also find the case where a line  $d_i$  matches a line  $d_j$  whose both end junctions  $I_j$  and  $J_j$  are valid and match junctions  $I_i$  and  $J_i$ , but only  $J_i$  belongs to  $d_i$ . See Fig. 4.5(a). It occurs mostly in Cases 1 or 2 when lines are matched around the cycle, after one match was established.

Table 4.1 shows us the different possibilities and appropriate actions taken for the first group and Table 4.2 does the same for the second group. In both cases it is assumed that the index in J with which we associate the line of connection is r. From the nature of the different configurations, r is always known. The problem is to determine

Table 4.1

Conditions for First-Group Junction Connections

	<p>I is a 2L junction and linked in indices r and s. Action: Extend line c to junction I and associate it with index t in I. I becomes a 3L junction, linked in all three indices. J is linked in index r.</p>
	<p>I is a 3L junction linked in indices r and s. Action: The line associated with index t in I is combined with line c. I becomes linked in three indices. J is linked in index r.</p>
	<p>I is a 3L junction linked in index s. Action: The average error of points on the line associated with r and t is calculated with respect to the natural extension of c. The line with the smaller average error is combined with c. I becomes linked in a second index. J becomes linked in r.</p>
	<p>I is a 1L junction linked in index s. Action: Extend line c to junction I and associate it with index t or r. I becomes a 2L junction, linked in two indices. J is linked in index r.</p>
	<p>I is a 2L junction linked in index r. Action: Calculate the average error of the points on the line associated with t, with respect to the natural extension of c. If the error is less than some threshold combine the two lines. I becomes linked in t and I in r.</p>
<p>Other</p>	<p>No Action</p>



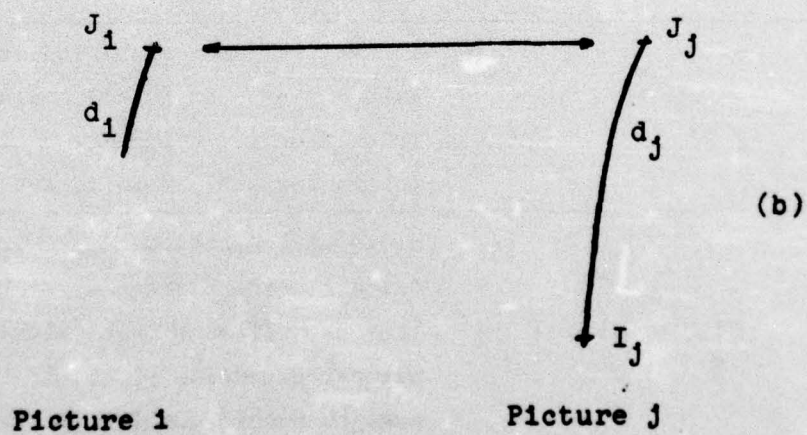
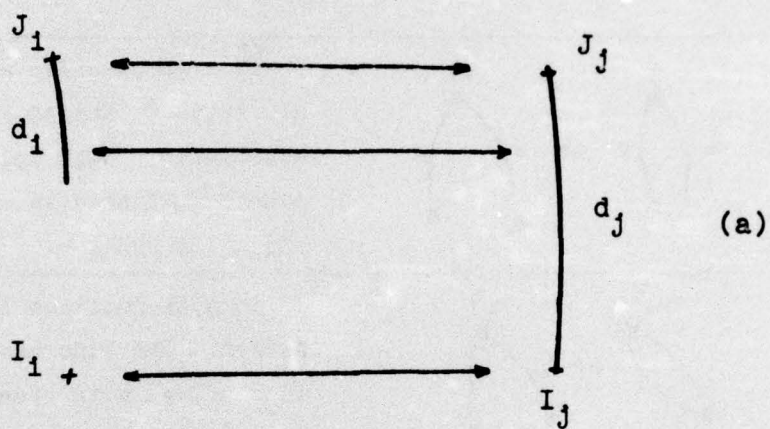


Fig. 4.5 Data Recovery Situation

with which index in  $I$  to associate this line. The possibility of having a 1L valid junction will be demonstrated later in this section. It is a result of certain data recovery actions.

By "empty" line mentioned twice in Table 4.2, we mean a line for which only the end points are known (the two junctions) and nothing is known about its nature. Its importance lies in creating the links, which in turn create new conditions for matching lines around  $I$  in the next pass. The linking of  $I$  and  $J$  may, for example, qualify junction  $I$  for analysis as one of the Case-1 configurations and may enable the application of Rules 1 and 2. We shall return to the detection and completion of missing connections once again in what we call second-level data recovery.

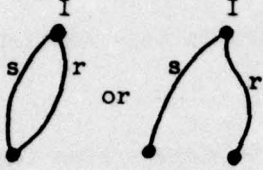
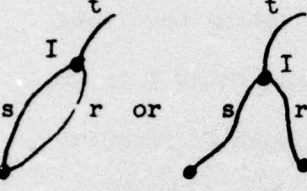
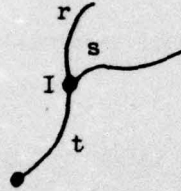
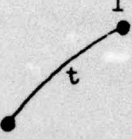
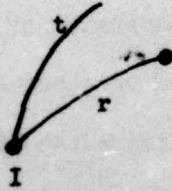
Let us consider the situation shown in Fig. 4.5(b). Lines  $d_i$  and  $d_j$  either match each other or are candidates for a match. But there is no match in picture  $i$  for junction  $I_j$ . This might be crucial for the matching decision for the lines  $d_i$  and  $d_j$  as we can see by comparing Fig. 4.3(b) with Fig. 4.3(c). If there is for  $I_j$  a matching junction  $I_k$  in the third picture, it could serve as a basis for further tests and data recovery. We may then have two different scenarios:

1. There exist a junction  $I_i$  in picture  $i$ , unmatched as yet, that is matchable to  $I_j$  and to  $I_k$ . If  $I_i$  lies on the natural extension of  $d_i$ , then  $(I_i, I_j, I_k)$  is declared as a match triple, and we conclude that the junctions  $J_i$  and  $I_i$  should be connected. We also match the lines  $d_i$  and  $d_j$  if they are not matched as yet.



TABLE 4.2

Conditions for Second-Group Junction Connections

	<p>I is a 2L junction linked in r and s. Action: Create a new "empty" line between I and J. Associate it with index t in I and index r in J. I becomes a 3L junction linked in all indices. The number of lines in J is increased by one and J is linked in index r.</p>
	<p>I is a 3L junction linked in 2 indices r and s. Action: Extend the line associated with t from I to J and associate it with index r in J. I is linked now in three indices. The number of lines in J is increased by 1. J is linked in index r.</p>
	<p>I is a 3L junction linked in index t. Action: Find which of the two lines associated with indices r and s has a natural extension to J. If there is one, extend the corresponding line to J. I is linked then in two indices. Number of lines in J is increased by 1, and it is linked in index r.</p>
	<p>I is a 1L junction linked in index t. Action: Create an "empty" line between I and J. Associate the line with index r in J and with one of the indices r or s in I. I becomes a 2L junction linked in two indices. The number of lines in J is increased by one, and J is linked in index r.</p>
	<p>I is a 2L junction linked in index r. Action: Check whether J is on the natural extension of the line associated with t. If so, extend this line to J. I is now linked in two indices. The number of lines in J is increased by 1 and it is linked in its index r.</p>
<p>Other</p>	<p>No Action</p>

2. If there is no junction  $I_i$  that is matchable to  $I_j$  and  $I_k$ , (and has not been matched as yet) then we build a synthetic junction.

This is done as follows: From  $I_j$  and  $I_k$  we can determine the approximate location in 3D of the corresponding vertex. It will be the point with the minimal sum of absolute distances from the projecting lines  $C_k I_k$  and  $C_j I_j$ . This point is found most easily by looking at the x, y coordinates of the intersection of the lines' projections on an x-y plane in a coordinate system whose z axis is normal to both lines. The calculated projection, from  $C_i$  on picture i, of this vertex will serve as the synthetic junction  $I_i$ . If the lines  $d_i$  and  $d_j$  are not matched as yet but the synthetic junction  $I_i$  lies on the natural extension of  $d_i$ , or if  $d_i$  and  $d_j$  already match, then we accept the synthetic junction as a new valid junction, declare the triple  $(I_i, I_j, I_k)$  as a match triple, and extend the line  $d_i$  to  $I_i$ .  $I_i$  is now a valid LL junction linked in one index.

To demonstrate how important a synthetic-junction formation can be for line matching, let us look at the three images shown in Fig. 4.6(a). Fig. 4.6(b) shows the line matching that can be established before the formation of the synthetic junction  $I_i$ . Fig. 4.6(c) shows the line matching established by the configuration of Fig. 4.1(i), possible only after  $I_i$  was formed.

Finally, an unordered junction for which Rules 1 and 2 cannot be applied, can be ordered if at least two of its lines are matched to two lines of an ordered junction so that the cyclic precedence of the first two will agree with that of the last two. The indices associated with the formerly unordered junction are then adjusted to comply with the cyclic order.



#### 4.5 Example

We shall describe some of the results obtained when a program for the foregoing procedure was tested on the scene shown in Fig's. 3.6, 3.7 and 3.8. In Fig. 4.7 we give a schematic description of the three pictures. First the unordered junctions  $6_1, 12_1, 17_1, 1_2, 3_2$ , and  $18_2$  were ordered each by applying Rule 1.

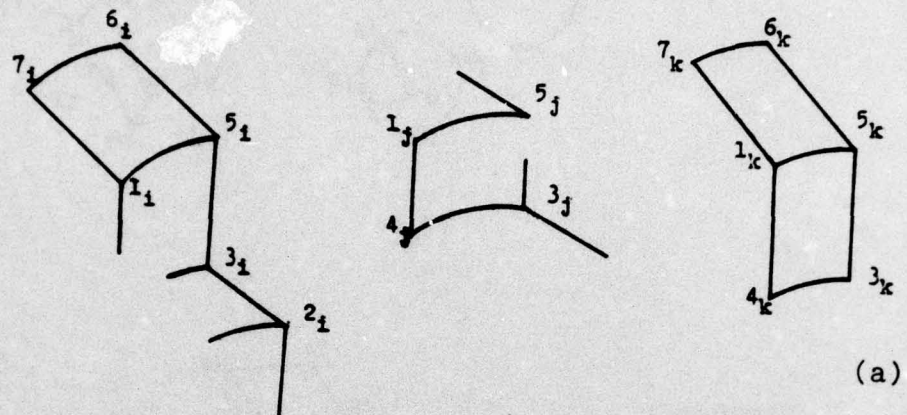
The configuration of Fig. 4.3(a) in  $14_1$  and  $3_2$  brought the match of lines  $\overline{12_1, 14_1}$  to  $\overline{4_2, 3_2}$  and  $\overline{13_1, 14_1}$  to  $\overline{3_2, 2_2}$ .

The configuration of Fig. 4.3(a) in  $14_1$  and  $17_3$  brought the match lines  $\overline{14_1, 13_1}$  to  $\overline{17_3, 18_3}$  and  $\overline{14_1, 26_1}$  to  $\overline{17_3, 16_3}$ . Combining this with the preceding match we get the match  $\overline{17_3, 18_3}$  to  $\overline{3_2, 2_2}$ . Through this match we are able also to arrange the cyclic order in junction  $17_3$ .

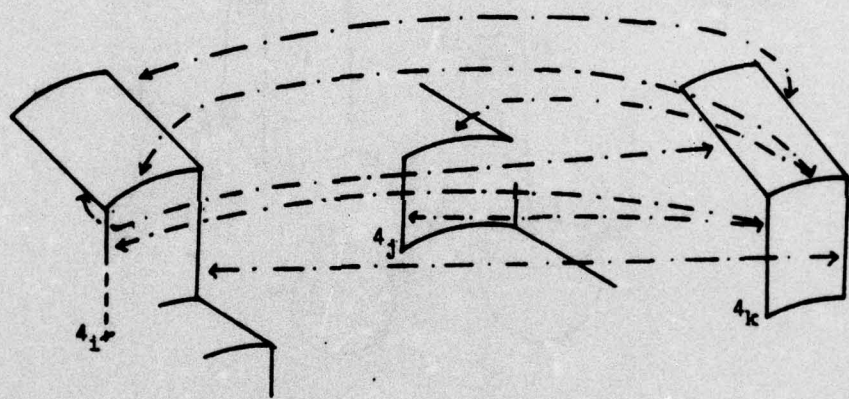
The configuration of Fig. 4.1(a) in  $11_1$  and  $9_2$  gave the match of lines  $\overline{11_1, 6_1}$  to  $\overline{9_2, 8_2}$ ,  $\overline{11_1, 13_1}$  to  $\overline{9_2, 2_2}$  and  $\overline{11_1, 12_1}$  to  $\overline{9_2, 4_2}$ .

The configuration of Fig. 4.3(a) in  $11_1$  and  $19_3$  gave the match of lines  $\overline{11_1, 6_1}$  to  $\overline{19_3, 20_3}$  which will match  $\overline{9_2, 8_2}$  - combining the above - and  $\overline{11_1, 13_1}$  to  $\overline{19_3, 18_3}$  which will match  $\overline{9_2, 6_2}$ . The result of the matching of the last group is that we can now establish the cyclic order in junction  $19_3$ .

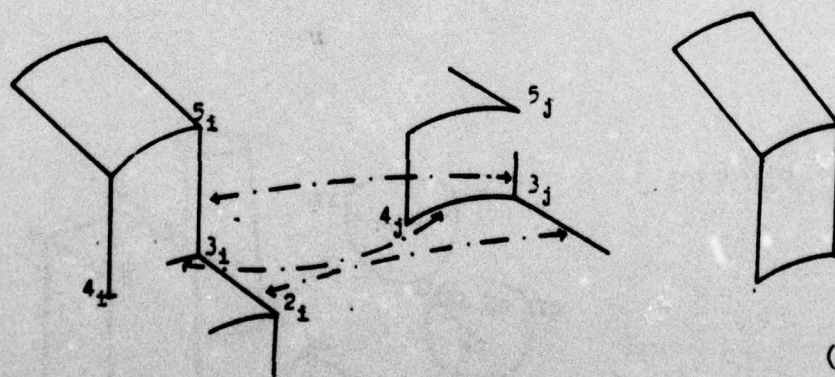
Also  $\overline{6_1, 9_1}$  match  $\overline{8_2, 1_2}$  based on the configuration of Fig. 4.3(a) in  $6_1$  and  $8_2$ ;  $\overline{6_1, 9_1}$  match  $\overline{20_3, 21_3}$  on basis of Fig. 4.3(a) in  $6_1$  and  $20_3$ ; and combining the two we get that  $\overline{20_3, 21_3}$  matches  $\overline{8_2, 1_2}$ .



(a)



(b)

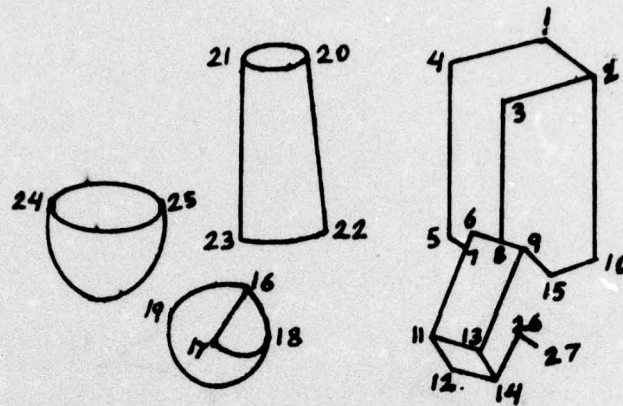


(c)

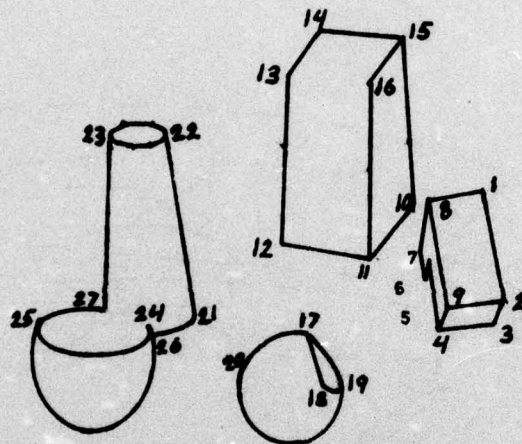
Fig. 4.6 Use of a Synthetic Junction



PICTURE 1



PICTURE 2



PICTURE 3

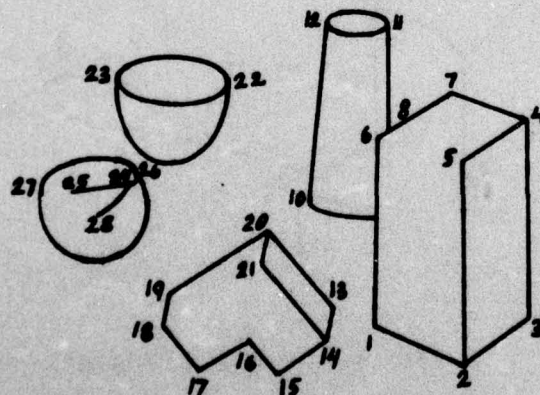


Fig. 4.7 Schematic Description of the Pictures Shown in Fig's. 3.6, 3.7 and 3.8.

From Fig. 4.1(a) in  $13_1$  and  $2_2$  we obtained the match of  $\overline{9_1, 13_1}$  to  $\overline{2_2, 1_2}$ . From Fig. 4.3(a) in  $13_1$  and  $18_3$  and a match between junctions  $9_1$  and  $21_3$  we obtain the linkage of junctions  $18_3$  and  $21_3$  by an "empty" line, which was matched to line  $\overline{13_1, 9_1}$  and thus to line  $\overline{2_2, 1_2}$ . Junction  $18_3$  was arranged cyclically by comparison with junction  $13_1$ , and this enabled the application of Rule 1 for ordering junction  $21_3$ ; this in turn enabled the application of Rule 1 for ordering junction  $13_3$ .

Next we obtained also the match of line  $\overline{9_1, 15_1}$  to  $\overline{21_3, 14_3}$  from the just formed Fig. 4.1(a) configuration in junctions  $9_1$  and  $21_3$ .

From 4.3(a) in  $1_1$  and  $4_3$  we found the match of  $\overline{1_1, 4_1}$  to  $\overline{4_3, 7_3}$ . The Fig. 4.1(a) configuration in  $2_1$  and  $15_2$  gave the match of  $\overline{2_1, 1_1}$  to  $\overline{15_2, 14_2}$ ,  $\overline{2_1, 10_1}$  to  $\overline{15_2, 10_2}$  and  $\overline{2_1, 3_1}$  to  $\overline{15_2, 16_2}$ . The configuration of Fig. 4.3(a) in  $2_1$  and  $5_3$ , together with the match of  $3_1$  and  $6_3$  brought the formation of an "empty" line between  $5_3$  and  $6_3$ , its match to line  $\overline{2_1, 3_1}$  (and thus to  $\overline{15_2, 16_2}$ ), the match of  $\overline{5_3, 2_3}$  to  $\overline{2_1, 10_1}$  (and thus to  $\overline{15_2, 10_2}$ ), the match of  $\overline{5_3, 4_3}$  to  $\overline{2_1, 1_1}$  (and thus to  $\overline{15_2, 14_2}$ ), and the cyclic ordering of junction  $5_3$ .

Line  $\overline{8_2, 7_2}$  was matched to  $\overline{20_3, 13_3}$  using the Fig. 4.1(a) configuration in  $8_2$  and  $20_3$ . Line  $\overline{11_2, 10_2}$  was matched to  $\overline{1_3, 2_3}$ , and  $\overline{11_2, 16_2}$  to  $\overline{1_3, 6_3}$  using the Fig. 4.3(a) configuration in  $11_2$  and  $1_3$ . Junction  $1_3$  is ordered by comparing it with junction  $11_2$ . This, together with the last paragraph results, permits the ordering of  $6_3$  by Rule 1. The configuration of Fig. 4.4(b) in  $13_2$  and  $7_3$  permits  $13_2$  and  $16_2$  to be connected by an "empty" line which is matched to line  $\overline{7_3, 6_3}$ ; line  $\overline{13_2, 14_2}$  is matched to  $\overline{7_3, 4_3}$  and thus also to line  $\overline{1_1, 4_1}$ .



In junctions  $4_1$  and  $13_2$  the configuration of Fig. 4.3(a) was formed only after the connecting of junctions  $13_2$  and  $16_2$ . Otherwise no qualified configuration could have been found there. This configuration brings the match of line  $\overline{4_1, 5_1}$  to line  $\overline{13_2, 12_2}$ .

We also tested for the configuration of Fig. 4.1(h) in junctions  $16_1$  and  $17_2$ . (In the beginning of the procedure we merge the two non-limb lines in every S junction into one line.) The extension of line  $\overline{17_2, 20_2}$  to junction  $18_2$  was rejected on the basis of condition\* 3. But a natural extension of line  $\overline{16_1, 19_1}$  to junction  $17_1$  was found, qualifying the configuration. Thus line  $\overline{16_1, 17_1}$  was matched to line  $\overline{17_2, 18_2}$ , line  $\overline{16_1, 19_1}$  was extended to junction  $17_1$  and matched to line  $\overline{17_2, 20_2}$ , which is now extended on the basis of the match to junction  $18_2$ . Finally, line  $\overline{16_1, 18_1, 17_1}$  was matched to line  $\overline{17_2, 19_2, 18_2}$ .

These matches were reached before making use of any synthetic junctions. Let us see how the matching ability is improved by them. As we can see, line  $\overline{3_1, 8_1}$  could not be matched. If the configuration found in  $3_1$  and  $6_3$  (Fig. 4.3(b)) could be modified to the configuration of Fig. 4.3(c) then the desired match could be established. (Junction  $4_1$  failed to be on the natural extension of  $\overline{3_1, 8_1}$ ). Hence, junctions  $11_2$  and  $1_3$  are used to create a synthetic junction in picture 1 which we denote as  $28_1$ . Junction  $28_1$  is found to be on the natural extension of  $\overline{3_1, 8_1}$  and thus is accepted as a valid junction. Line  $\overline{3_1, 8_1}$  is extended

---

\* See discussion concerning natural extension in Section 4.3

to be line  $\overline{3_1, 28_1}$  and the triple  $(28_1, 11_2, 1_3)$  is declared to be a match triple.

The configuration of junctions  $11_2$  and  $1_3$  force the connections of junction  $28_1$ . Thus junction  $28_1$  should be connected to junctions  $10_1$  and  $5_1$  under the rules of Table 4.2. It is found to be on the natural extension of  $\overline{10_1, 15_1}$  which is thus extended to be line  $\overline{10_1, 28_1}$ . It is also found to be on the natural extension of line  $\overline{5_1, 7_1}$  which turns to be line  $\overline{5_1, 28_1}$ . The new configuration created now in junctions  $3_1$  and  $6_3$  (of form Fig. 4.3(e)), causes the match of  $\overline{3_1, 28_1}$  to  $\overline{6_3, 1_3}$  and thus also to  $\overline{16_2, 11_2}$ , as well as the connection with an "empty" line of  $3_1$  to  $4_1$  (and its match to  $\overline{6_3, 7_3}$  and thus also to the "empty" line  $\overline{13_2, 16_2}$ ). We also get the match of  $\overline{10_1, 28_1}$  to  $\overline{1_3, 2_3}$  (configuration of Fig. 4.3(a) in  $28_1$  and  $1_3$ ) and thus also to  $\overline{10_2, 11_2}$ . In addition we get the match of  $\overline{5_1, 28_1}$  to  $\overline{12_2, 11_2}$  (configuration of Fig. 4.1(a) in  $28_1$  and  $11_2$ ), and finally the following junctions are ordered cyclically as the result of all this additional matching and linking:  $1_1, 3_1, 4_1, 5_1, 10_1, 28_1, 10_2, 12_2, 13_2, 14_2, 16_2$  and  $7_3$ .

#### 4.6 Matching Lines by Range

The technique of matching lines around matched junctions cannot be applied when the lines are closed, when they have end junctions that are both of type A, when the ends are both open, or when one end junction is of type A and the other end is open. Additional tools are needed for establishing a match in these situations. We define for a line  $g_1$  in picture  $i$  range limits with respect to picture  $j$  as the two straight lines,



with the maximum angle of opening, going through  $C_{ij}$  and having points in common with  $g_i$ . (Lines a and b in Fig. 4.8). For a closed line the range limits are the two tangents to the line from  $C_{ij}$ .

Taking two pictures i and j, we create in each picture, for all qualified lines, pairs of range limits with respect to the other picture. If we now project both pictures on a plane  $P_q$  (as explained in Chapter III and shown in Fig. 3.2) we get all the range limits from both pictures as a set of lines going through  $C_{ijq}$ . If we take the range limits of a line  $g_i$ , then when projected on  $P_q$ , they form an angle  $\alpha$ . The range limits of the line  $h_j$  when reprojected on  $P_q$  form the angle  $\beta$ . We define a ratio  $R_t = \frac{\gamma}{\alpha + \beta}$  where  $\gamma$  is the overlap angle (See Fig. 4.8). When  $R_t$  is bigger than some threshold, we say that  $g_i$  and  $h_j$  are matchable by range (but are not matched yet). When we have three lines in three pictures, lines  $g_i, h_j$  and  $f_n$ , pairwise matchable by range, then we say that the lines are matched by range. In our example the lines  $\overline{23_1}, \overline{22_1}, \overline{21_2}, \overline{26_2}$  and  $\overline{10_3}, \overline{9_3}$ , among others, are matched by range.

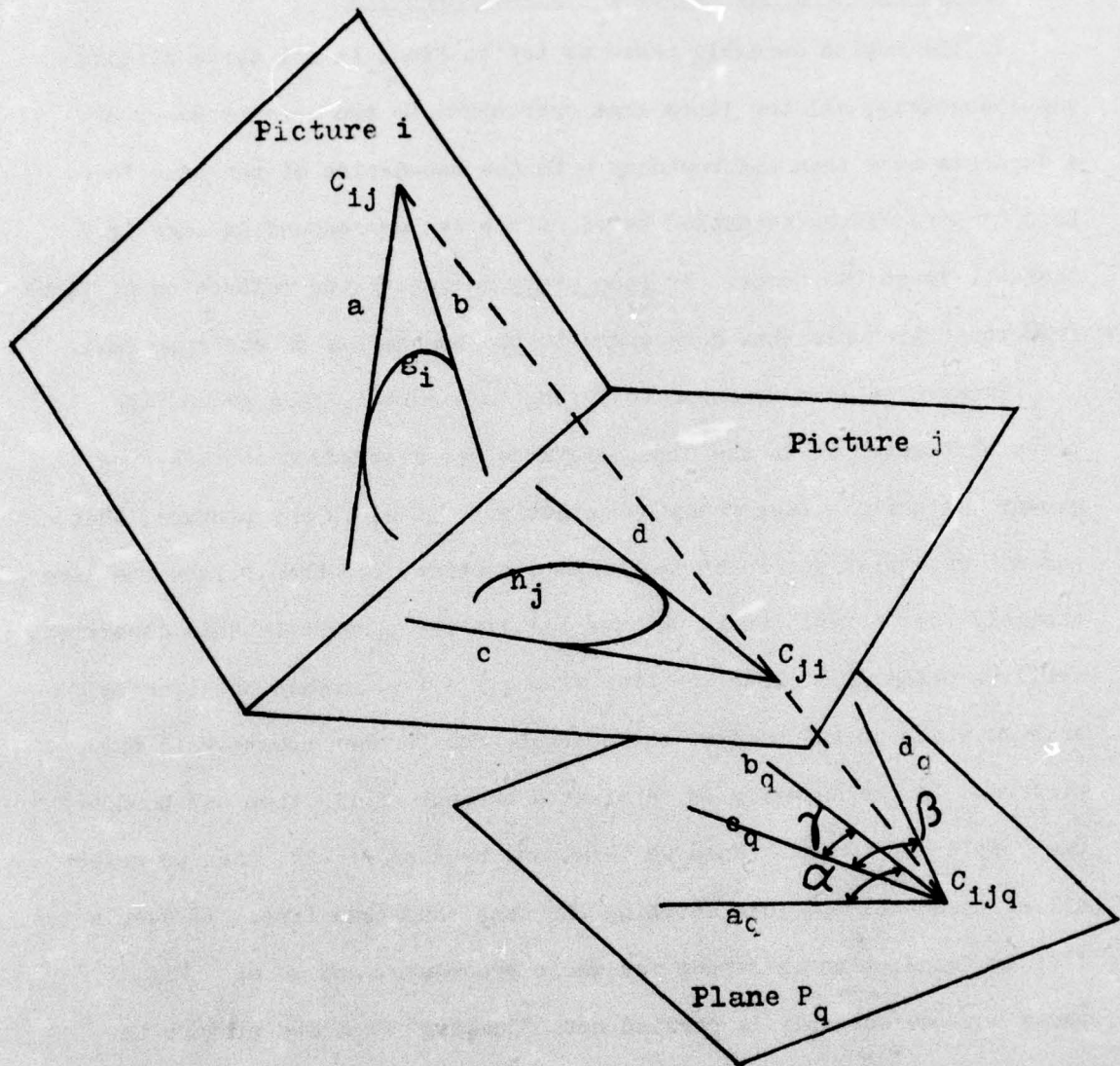


Fig. 4.8 Matching Lines by Range

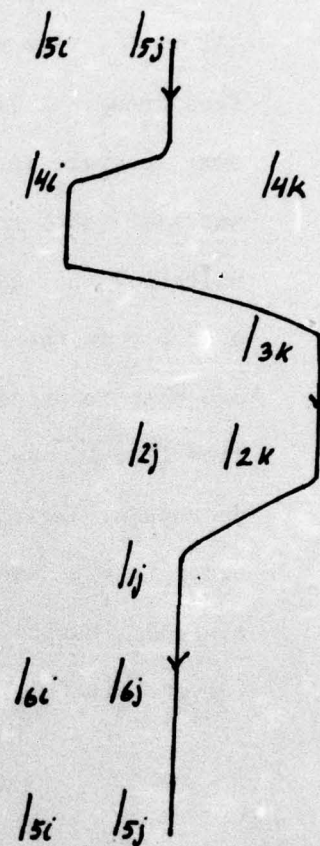
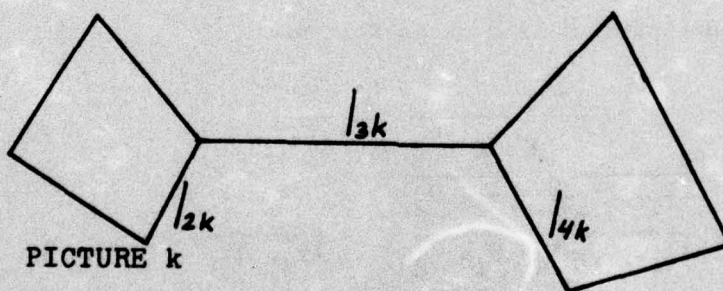
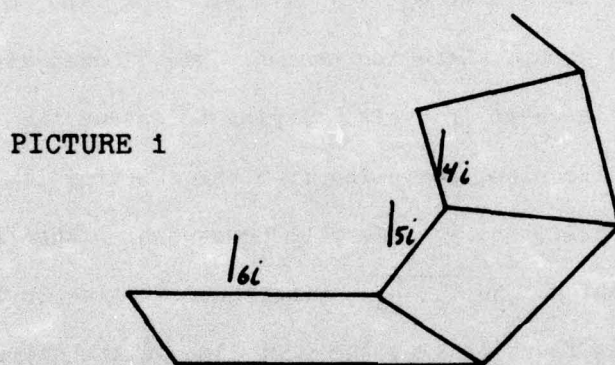
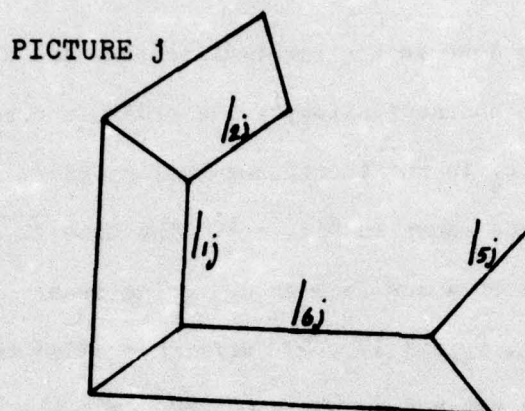


## V. FINAL DESCRIPTION

### 5.1 Region Assembly and Second-Level Data Recovery

In the region assembly phase we try to find, in all three pictures simultaneously, all the lines that correspond to the same boundary or - if a face has more than one boundary - to the boundaries of the same face. Each line should be assembled twice, since its corresponding edge is shared between two faces. By face group we denote the collection of lines from three pictures that correspond to the boundaries of the same face.

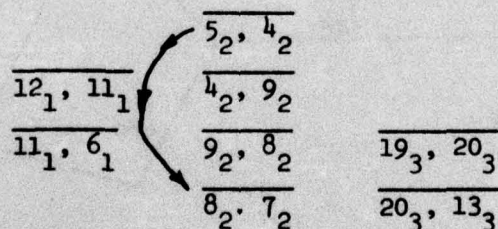
Whenever a line is found to belong to a certain face group, the lines that match it in the other pictures are also added to this face group. We build a face group by selecting a line, in any picture, that was not assembled yet in at least one direction. We then follow the line assembly (see definition in Chapter II), in the picture in this direction, until we either (1) close the line assembly - i.e., reach the line we started with, or (2) no line can be found for further assembly in this picture. If the assembly is terminated because of (1) then one boundary was completely traced. When we terminate because of (2), then we select a line in another picture matching the last assembled line. If such a line is found, then we repeat the whole procedure, and so on. Fig. 5.1 shows how the assembly is carried out, "jumping" from one picture to another. If the assembly process is blocked in all three pictures, we return to the starting point and trace the lines in the opposite direction, changing lines at the junctions in decreasing cyclic order and following the above-described procedure until we are blocked again in all three pictures. Although we are now tracing the lines in the opposite direction,





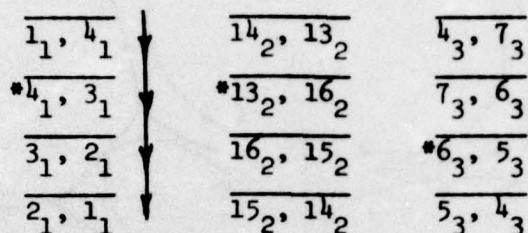
we record the trace as if it were done in the right direction, and a change from line  $c_i$  to line  $d_i$  in the decreasing cyclic order is recorded as a change from line  $d_i$  to line  $c_i$  in the increasing cyclic order.

For example consider the scene shown in Fig. 4.7. The line  $\overline{11_1, 6_1}$  was selected as the first element of a new face group, going from  $11_1$  to  $6_1$ . Its matching lines  $\overline{9_2, 8_2}$  and  $\overline{19_3, 20_3}$  were then added to the face group. A line next in cycle was not found in  $6_1$  and hence the line next in cycle in  $8_2$  was picked - line  $\overline{8_2, 7_2}$ . Its matching line  $\overline{20_3, 13_3}$  was also added to the face group. Here the assembly was blocked since neither  $7_2$  nor  $13_3$  had a line next in cycle. Trying to extend the assembly in the opposite direction, beginning from the starting lines, we are able to proceed in both  $11_1$  and  $9_2$ . Next to be assembled then was line  $\overline{11_1, 12_1}$  and its matching line  $\overline{9_2, 4_2}$ . Since no next line in the decreasing cyclic order was found in  $12_1$ , the next line in the decreasing order in  $4_2$ , namely lines  $\overline{4_2, 5_2}$ , was picked. The assembly terminates here since  $5_2$ , having no match, was not validated and not ordered. The recording of this face group assembly is done as follows:



Let us consider another example, in which a closed boundary is completely traced. The first line picked for this face group is  $\overline{1_1, 4_1}$ .

(The stars indicate the places where the linkage between the two junctions is the result of the data recovery phase):



The existence of a virtual junction usually indicates the existence of a multiple-boundaries face. Thus on assembling the lines that belong to a single boundary, all S and A junctions are remembered\*; when the boundary trace is terminated, we return to these junctions, search for the other end of its limb, and start there another line assembly for the same face group. The limbs themselves are also added to this face group.

When all lines have been twice assembled, we may try some further data recovery actions. This second-level data recovery is carried out in the three following configurations shown in Fig. 5.2.

1. Two lines  $l$  and  $k$  in the same picture, with one non-valid end junction in each, are assembled into two face groups  $M$  and  $N$ ,  $k, l \in M$  and  $k, l \in N$ . Action: Combine  $l$  and  $k$  into one line.
2. A line  $l$  with one non-valid end junction is assembled into two face groups  $M$  and  $N$ , and there is in the same picture a 2-lines junction  $J$  with lines  $m$  and  $n$ ,  $m \in M$  and  $n \in N$ . Action: Extend the line  $l$  to junction  $J$ .

---

\* In Chapter III it was stated that the two non-limb lines in every S junction are merged into one line. Thus, on assembling the lines, there is no change of lines in an S junction; the junction is bypassed and remembered.



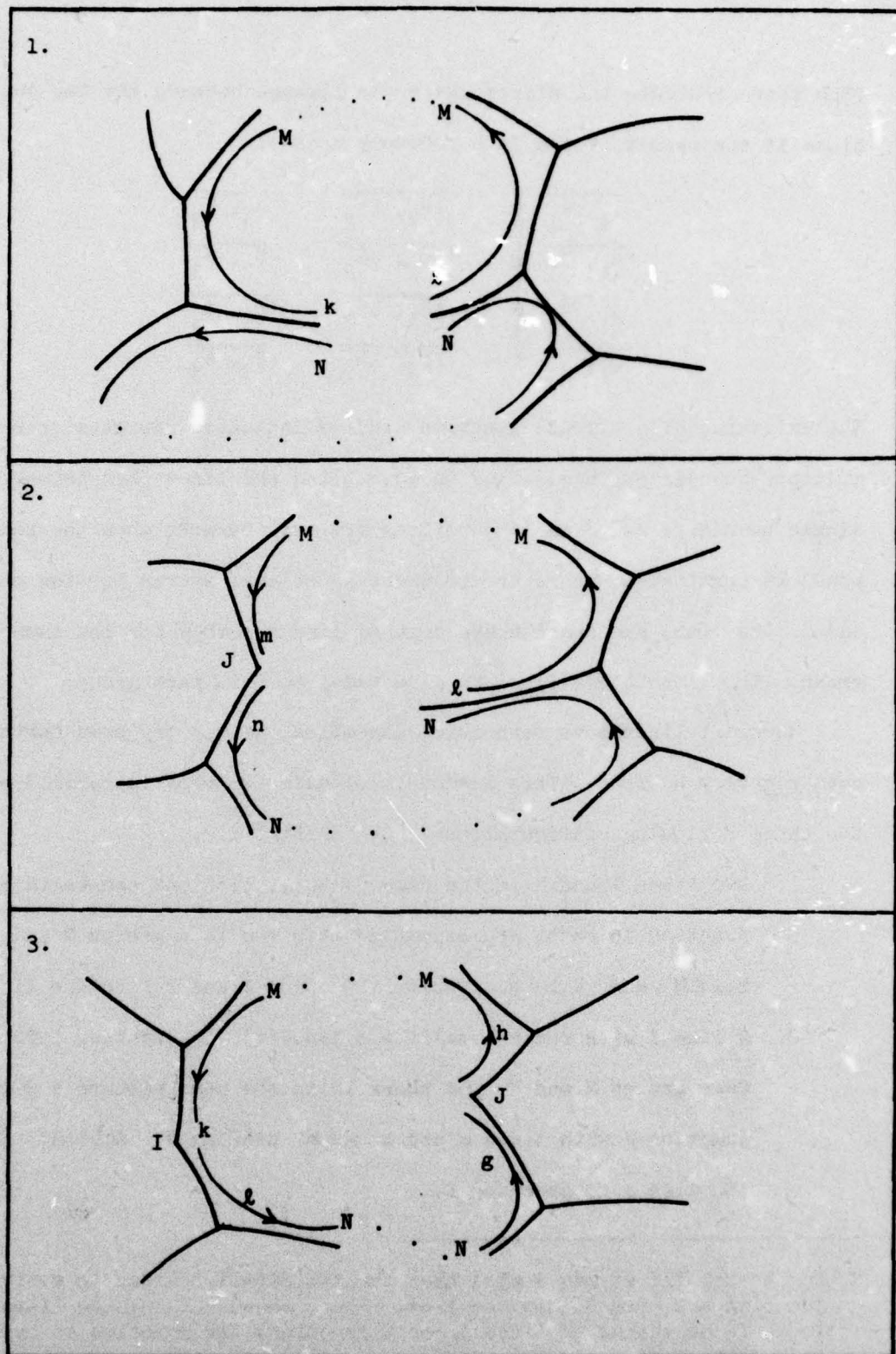


Fig. 5.2 Three Configurations for Second-Level Data Recovery

3. There are two 2-lines junctions I and J in the same picture with lines k,l and h,g, respectively, such that k and h are assembled into face group M, and l and g are assembled into face group N. Action: Create an "empty" line between the two junctions I and J.

The data recovery actions utilize the fact that two components assembled simultaneously into two face groups must correspond to the intersection of the two corresponding faces. There is of course the possibility that this intersection is sectioned into several separate edges and that the two components correspond to two different sections; but at this stage of the recognition phase, with no evidence to the contrary, we assume that they belong to the same section.

After any second-level data recovery actions have been performed, we try again to match lines, using the new conditions. Then conditions for second-level data recovery are checked again, and so on until no further line matching is possible. The face-group set is updated each time new results are obtained.

## 5.2 Object Formation

Each of the face groups defines (corresponds to) a face of one of the bodies in the scene. After the face-group assembly and second-level data recovery is completed, we are ready for the formation of objects. Each object is a set of face groups, with no common elements between two sets. Every face group must belong to some set. A new object is formed by picking a face group that is still free, then recursively adding to the set every face group, not as yet in the set, which has a line in common with any of the face groups already in the set.



### 5.3 Type of Face Determination

We would like to know for each face group whether it defines a planar or curved face. Toward that end let us define the predicate  $C(A)$  for every face group  $A$  as "the face defined by  $A$  is curved". Whenever a face group has a limb in it, we have, by definition,  $C(A)=1$ . When two face groups  $A$  and  $B$  share a straight line\*, no information is conveyed about the faces' nature. However, when they share a curved line, we must have  $C(A) + C(B) = 1$ , that is, one of the faces or both must be curved. Thus, we have for every object a set of logical equations formalizing the constraints imposed by the types of lines.

The philosophy underlying our strategy for determining the nature of faces is the following: Given several pictures, every curved face will somehow manifest itself or there is no way to recognize it as curved. Thus, we are looking for a minimal cover to satisfy the set of logical equations for every object based on the information collected from all three (or more) pictures. This minimal cover will determine which of the face groups define a curved face.

We find the minimal cover by means of a minimum-cost tree search, where the cost function is the number of faces thus far determined as curved. Consider the one-picture example shown in Fig. 5.3. The set

---

\* In Chapter IV we discussed how we determine whether a line is straight or curved. We say that two face groups share a curved line, if in at least one picture there is a curved line which they share.

of equations formalizing the lines constraints is:

1.  $C(B)=1$
- \*2.  $C(B)+C(A)=1$
- \*3.  $C(B)+C(F)=1$
- \*4.  $C(B)+C(E)=1$
5.  $C(C)+C(E)=1$
6.  $C(A)+C(C)=1$
- \*7.  $C(B)+C(D)=1$
8.  $C(F)+C(C)=1$
9.  $C(C)+C(D)=1$
- \*10.  $C(G)+C(B)=1$

The tree search, beginning with the first equation, is given in Fig. 5.4. The starred (\*) equations are those that give no additional constraint since one of their elements participates in a single-member equation (e.g., B). The minimum-cost tree search gave the cover that determined the faces corresponding to B and C as curved. (Which agrees with our intuitive interpretation of the picture). After the nature of the faces is determined we may wish to classify those that are curved as to the type of their quadric equation. The flowchart of Fig. 5.5 shows a way of doing this.

#### 5.4 Face Equations

We have already explained how the approximate 3D location of a point is determined from its two projections by taking the point of minimum sum of distances from the two projecting lines. If we are given three



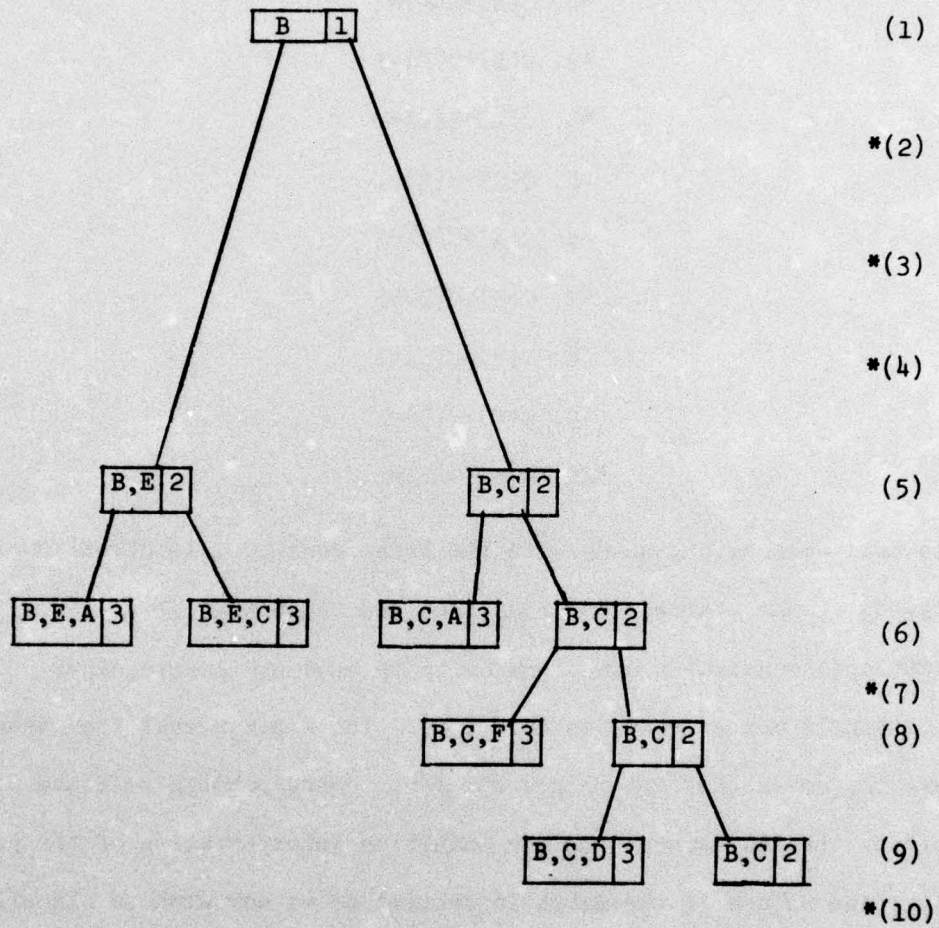
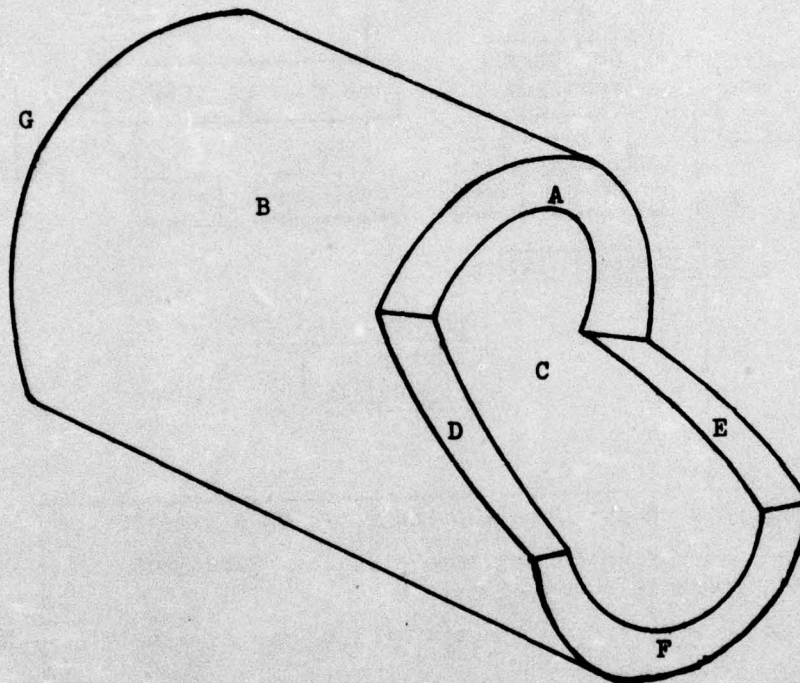
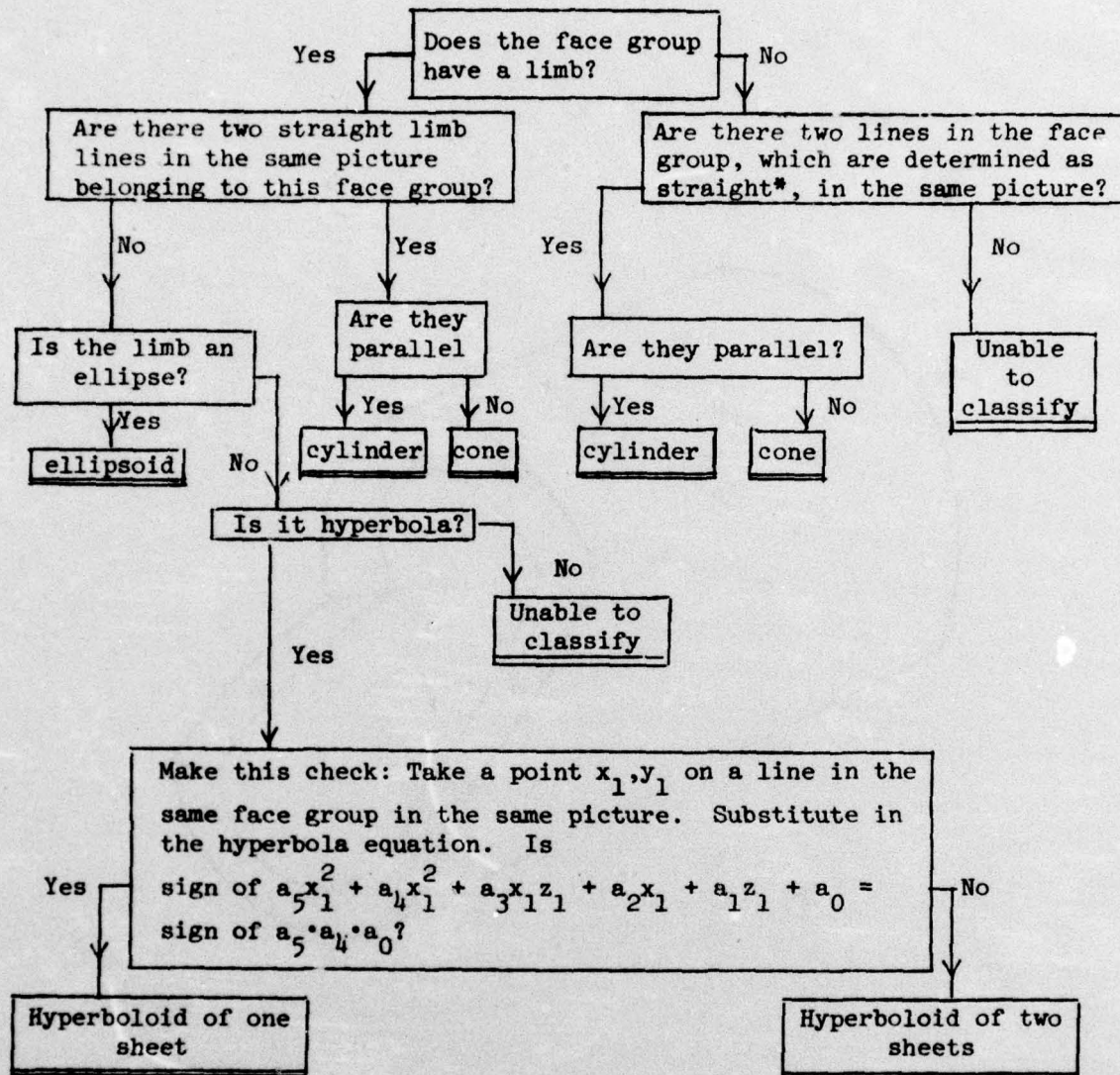


Fig. 5.3 Tree Search for Minimal Cover to Determine Curved-Face Group. (The number of the equation yielding each particular new level of nodes is indicated at the right.)



**Fig. 5.4** Curved-Face Determination (Note that Face Group B will Consist of Two Line Assemblies and Two Limbs)





\* Note: straight lines exist also on other types of quadric surfaces but we assume the most likely possibility. It is also assumed that parallelization is not badly distorted.

Fig. 5.5 Flow Chart for Determination of Type of Quadric Surface

projections of a point, then for a better approximation we may determine three 3D locations (by taking two projecting lines at a time) and then take their mean. This will give a very good approximation of the 3D location.

If we wish to have the equation of a face which we know to be planar, all its vertices which have at least two matched images can be collected in 3D and a plane equation fitted to them by minimizing the sum of square errors.

We shall now show how the equation of a curved face can be determined from its limb line in three different pictures.

The equation of a quadric surface is

$$Q(x,y,z) \equiv \underline{u}^T \cdot \underline{A} \cdot \underline{u} = 0$$

where A is the matrix of coefficients

$$\underline{A} = \begin{bmatrix} a_{11} & \frac{1}{2}a_{12} & \frac{1}{2}a_{13} & \frac{1}{2}a_{14} \\ \frac{1}{2}a_{12} & a_{22} & \frac{1}{2}a_{23} & \frac{1}{2}a_{24} \\ \frac{1}{2}a_{13} & \frac{1}{2}a_{23} & a_{33} & \frac{1}{2}a_{34} \\ \frac{1}{2}a_{14} & \frac{1}{2}a_{24} & \frac{1}{2}a_{34} & a_{44} \end{bmatrix}$$

and u is a point in homogeneous coordinates

$$\underline{u} = (x,y,z,1)^T$$

If we denote by  $u_i$  the center of projection for picture i, then the limb of Q with respect to  $u_i$  lies on the polar plane of  $u_i$  with respect to Q.

$$P(x,y,z) \equiv \underline{u}_i^T \cdot \underline{A} \cdot \underline{u} = 0$$



The cone of projection belongs to the linear family

$$\underline{u}^T \cdot \underline{A} \cdot \underline{u} - k(\underline{u}_i^T \cdot \underline{A} \cdot \underline{u})^2 = 0$$

Since  $\underline{u}_i$  belongs to the cone of projection, we get

$$k = \frac{1}{\underline{u}_i^T \cdot \underline{A} \cdot \underline{u}_i}$$

and the cone of projection for the surface onto picture i is

$$c(x,y,z) \equiv \underline{u}_i^T \cdot \underline{A} \cdot \underline{u}_i \cdot \underline{u}^T \cdot \underline{A} \cdot \underline{u} - (\underline{u}_i^T \cdot \underline{A} \cdot \underline{u})^2 = 0$$

Expressing this cone in picture i's coordinate system and substituting 0 for  $y^i$  coordinate, we get the equation of the limb expressed in terms of the unknown quadric's coefficients. The coefficients of the calculated limb line, which are functions of the desired coefficients  $a_{n,j}$ , should be proportional to the coefficients of the fitted conic. This gives us, for ideal data, six non-linear equations in the ten unknowns plus the unknown proportionality coefficient:

$$F_{n,i}(a_{11}, a_{12}, \dots, a_{33}, a_{44}, c_i) = 0$$

$$n=1 \dots 6$$

A second picture will give us six more equations but only four of them will be independent of the first picture; it will add, however, an unknown proportionality coefficient.

$$F_{n,j}(a_{11}, a_{12}, \dots, a_{33}, a_{44}, c_j) = 0$$

$$n=1 \dots 6$$

The third picture will add six more equations and another unknown

$$F_{n,k}(a_{11}, a_{12}, \dots, a_{33}, a_{44}, c_k) = 0$$

$$n=1 \dots 6$$

but only two of the equations will be independent of the other two pictures. This, together with the normalization condition

$$\sum_{j=1}^4 \sum_{i=1}^4 a_{ij}^2 = 1$$

give us thirteen non-linear equations in thirteen unknowns. In the case of real data we can solve for the unknowns by looking for a minimum for the functions  $F$  together with the last equality equation. The algorithms concerning the faces' equations were not programmed.



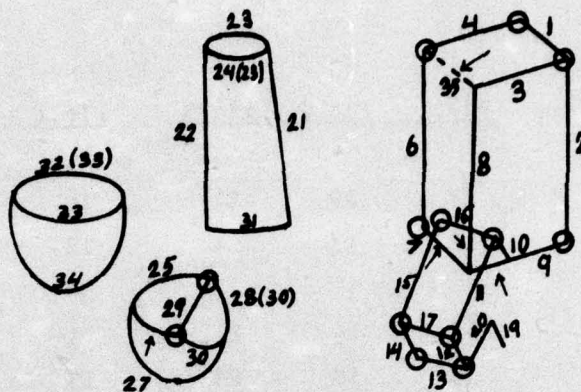
## VI. EXPERIMENTAL RESULTS

### 6.1 First Example

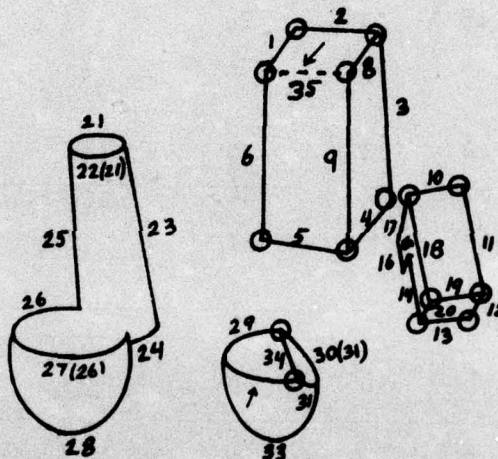
For the first example consider the scene which was discussed in Chapters 3 and 4 (see Fig. 4.7). The face groups reported by the program are as follows: (For line notation see Fig. 6.1, where also part of the data recovery results are shown.)

	<u>Pic.1</u>	<u>Pic.2</u>	<u>Pic.3</u>	<u>Pic.1</u>	<u>Pic.2</u>	<u>Pic.3</u>
<u>Face Group 1</u>						
Lines:	33	26	28			
Limb Lines:	34	28	26			
<u>Face Group 2</u>						
Lines:	33	26	28			
<u>Face Group 3</u>						
Lines:	31	24	10	23	21	13
Limb lines:	22,21	23,25	14,11			
<u>Face Group 4</u>						
Lines:	31	24	10			
<u>Face Group 5</u>						
Lines:	30	31	-			
	25	29	-			
Limb lines:	27	33				
<u>Face Group 6</u>						
Lines:	30	31	-	29	34	-
<u>Face Group 7</u>						
Lines:	29	34	-	25	29	-
<u>Face Group 8</u>						
Lines:	23	21	13			
<u>Face Group 9</u>						
Lines:	20	-	19	13	13	-
	-	14	-			

PICTURE 1



PICTURE 2



PICTURE 3

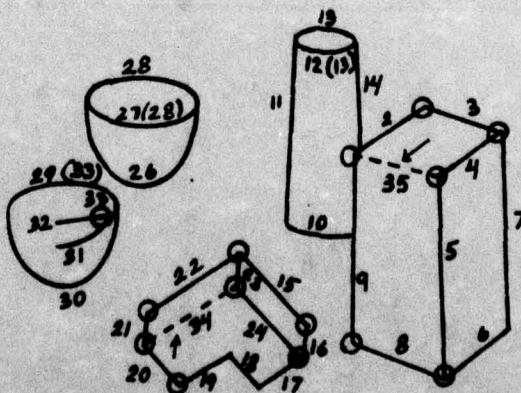


Fig. 6.1 First Example. (The Circled Junctions are Cyclically Arranged. The Arrows Point at Data Recovery Results. The Broken Lines Indicate "Empty Lines", for which only the End Points are Known.)



	<u>Pic.1</u>	<u>Pic.2</u>	<u>Pic.3</u>	<u>Pic.1</u>	<u>Pic.2</u>	<u>Pic.3</u>
<u>Face Group 10</u>						
Lines:	17	19	21	14	20	-
	13	13	-	12	12	20
<u>Face Group 11</u>						
Lines:	17	19	21	11	11	34
	16	10	23	15	18	22
<u>Face Group 12</u>						
Lines:	16	10	23	10	-	24
	-	-	16	-	17	15
<u>Face Group 13</u>						
Lines:	15	18	22	-	17	15
	14	20	-	-	14	-
<u>Face Group 14</u>						
Lines:	12	12	20	20	-	19
	11	11	34	10	-	24
	-	-	17			
<u>Face Group 15</u>						
Lines:	9	4	8	7	5	-
	-	-	6			
<u>Face Group 16</u>						
Lines:	8	9	9	35	35	2
	6	6	-	7	5	-
<u>Face Group 17</u>						
Lines:	8	9	9	9	4	8
	2	3	5	3	8	35
<u>Face Group 18</u>						
Lines:	6	6	-	4	1	3
	-	-	7			
<u>Face Group 19</u>						
Lines:	4	1	3	35	35	2
	3	8	35	1	2	4
<u>Face Group 20</u>						
Lines:	2	3	5	-	-	6
	1	2	4	-	-	7

The following face groups were labelled as curved: 1, 3, 5. The program determined the following bodies:

Body 1 with face groups 1 and 2.

Body 2 with face groups 3, 4 and 8.

Body 3 with face groups 5, 6 and 7.

Body 4 with face groups 9, 10, 11, 12, 13 and 14.

Body 5 with face groups 15, 16, 17, 18, 19 and 20.

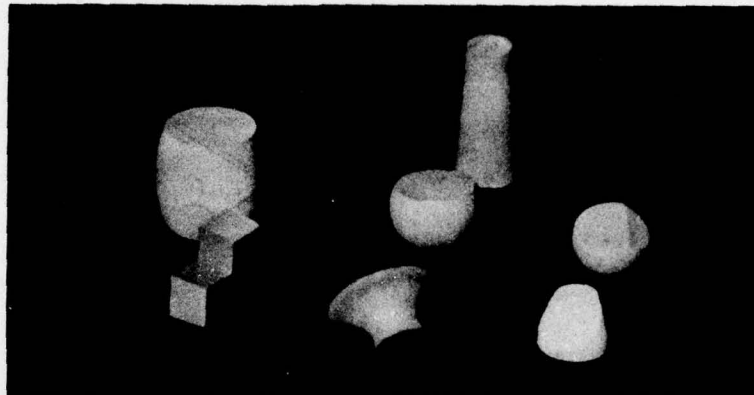
## 6.2 Second Example

The scene analyzed for the second example is shown in Fig. 6.2. The input data are shown in Figures 6.3, 6.4, and 6.5. Part of the data recovery results is shown in Fig. 6.6. The computer output is given in Fig. 6.7 (a) through (f).

## 6.3 Third Example

The scene for the third example is shown in Fig. 6.8. Input data is shown in Fig's. 6.9, 6.10, and 6.11. Part of the data recovery results is shown in Fig. 6.12. The computer output is given in Fig. 6.13 (a) through (f). Of special interest is the determination of the curved faces.

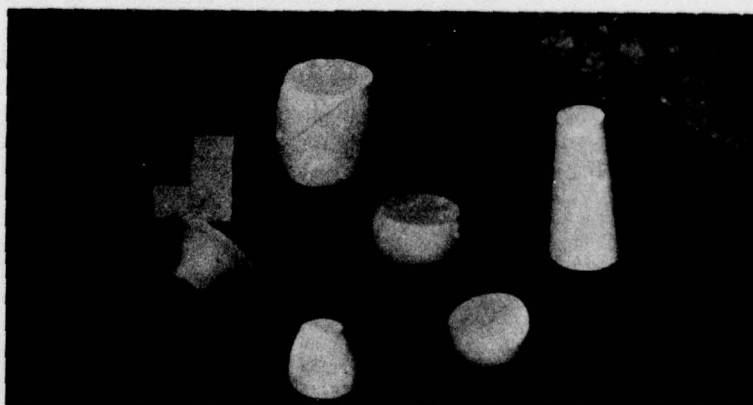




(a)



(b)



(c)

Fig. 6.2 Second Scene Example. (a) Picture 1,  
(b) Picture 2, (c) Picture 3

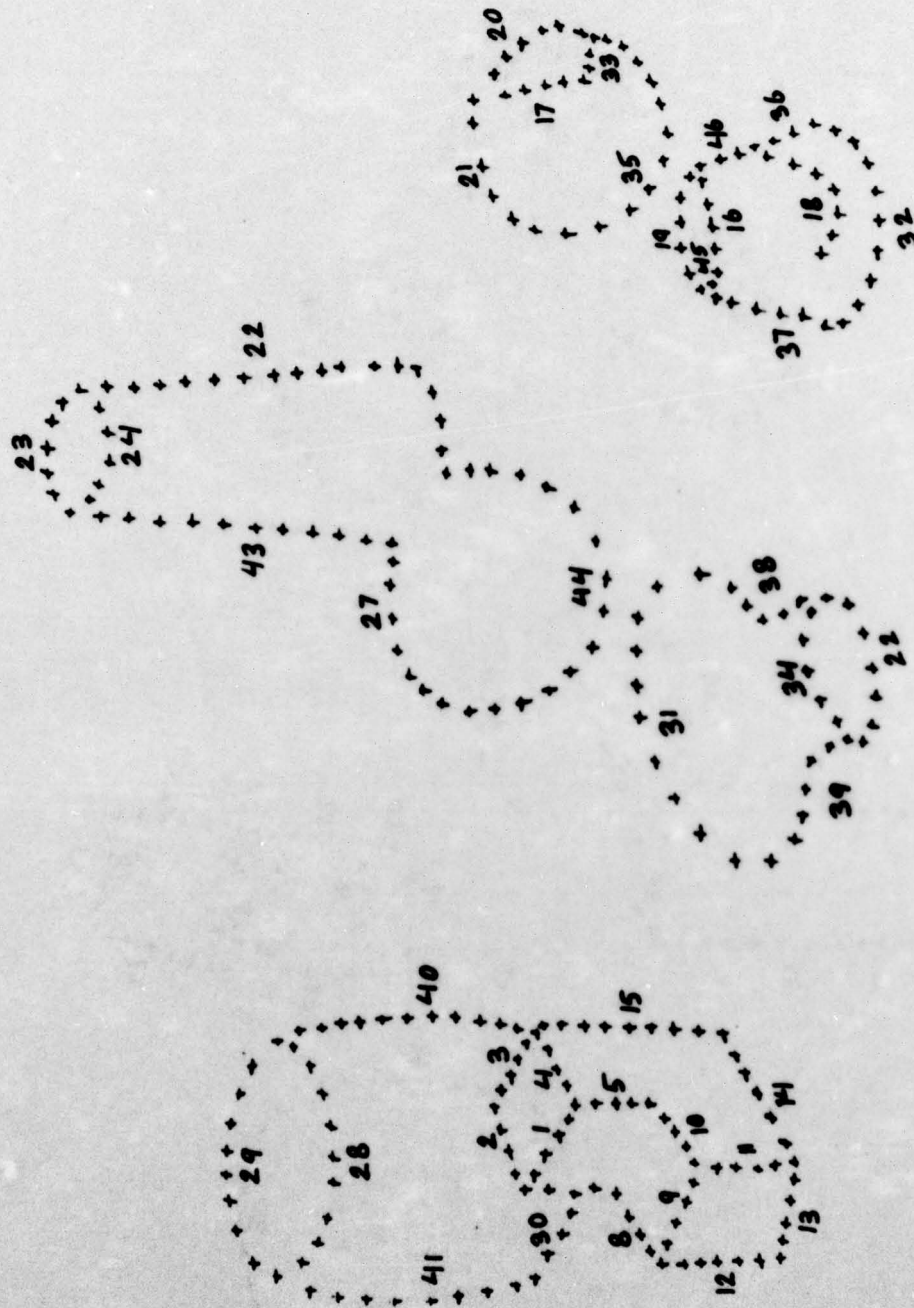


Fig. 6.3 Second Scene: Picture-1 Input Data



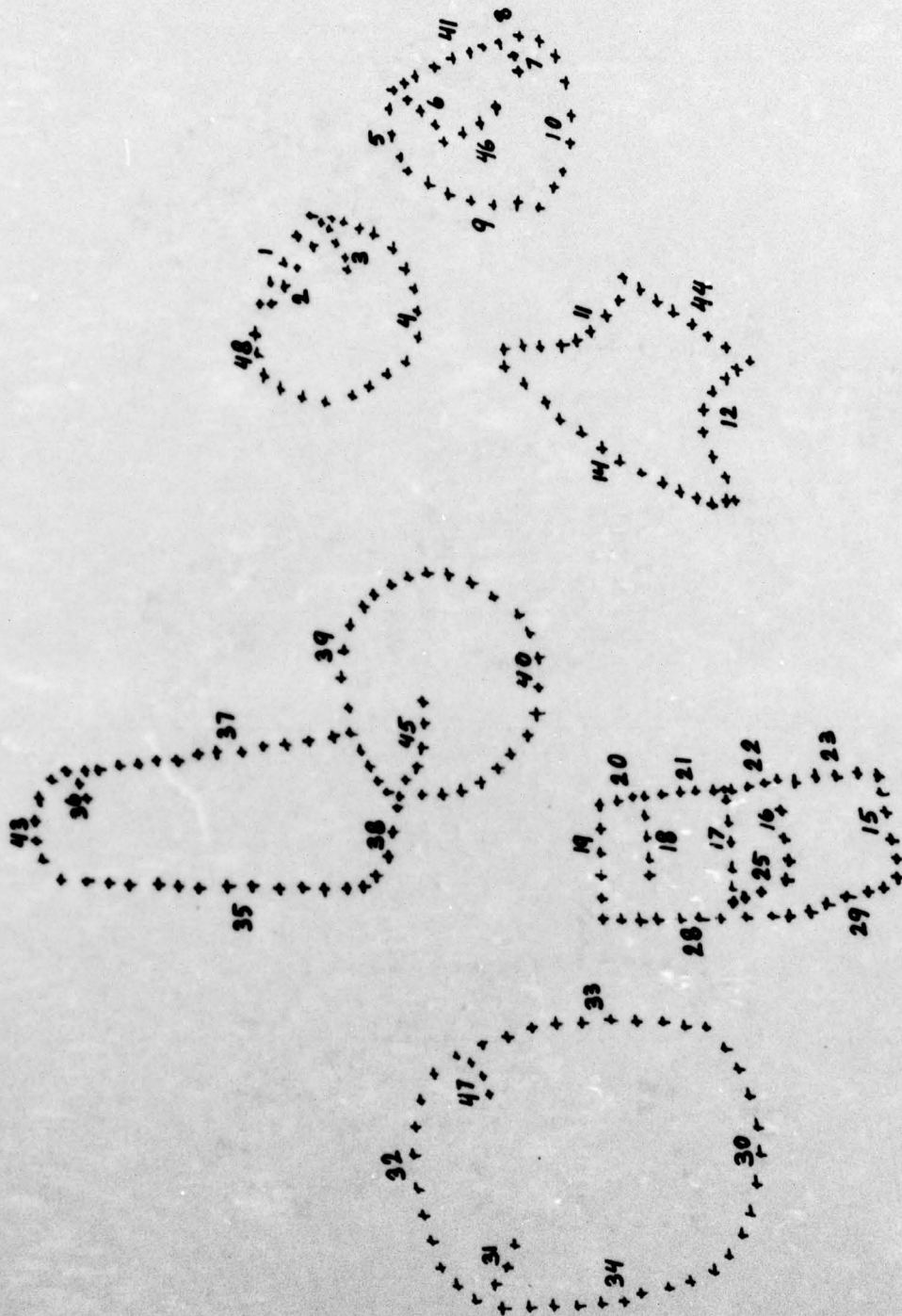


Fig. 6.4 Second Scene: Picture-2 Input Data

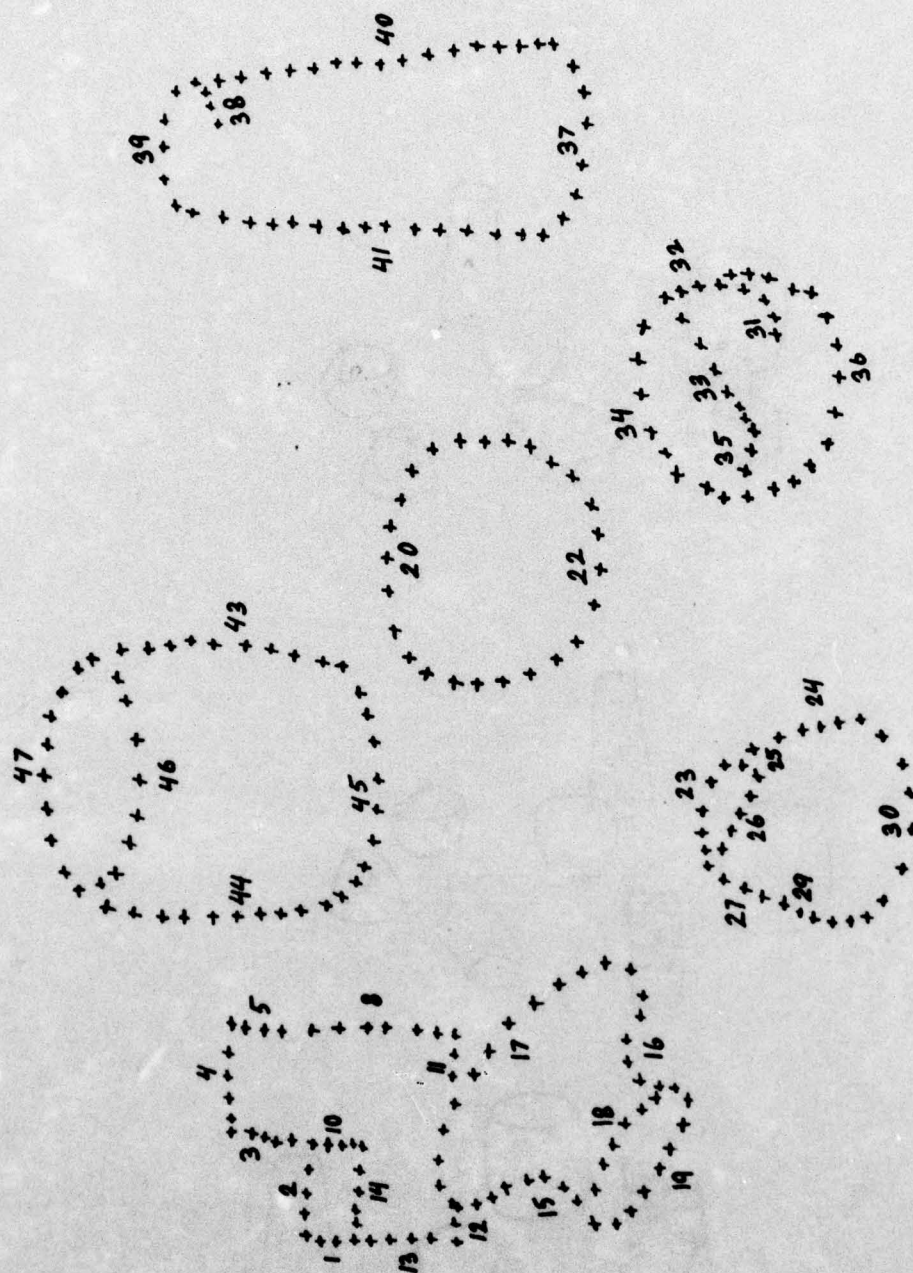


Fig. 6.5 Second Scene: Picture-3 Input Data



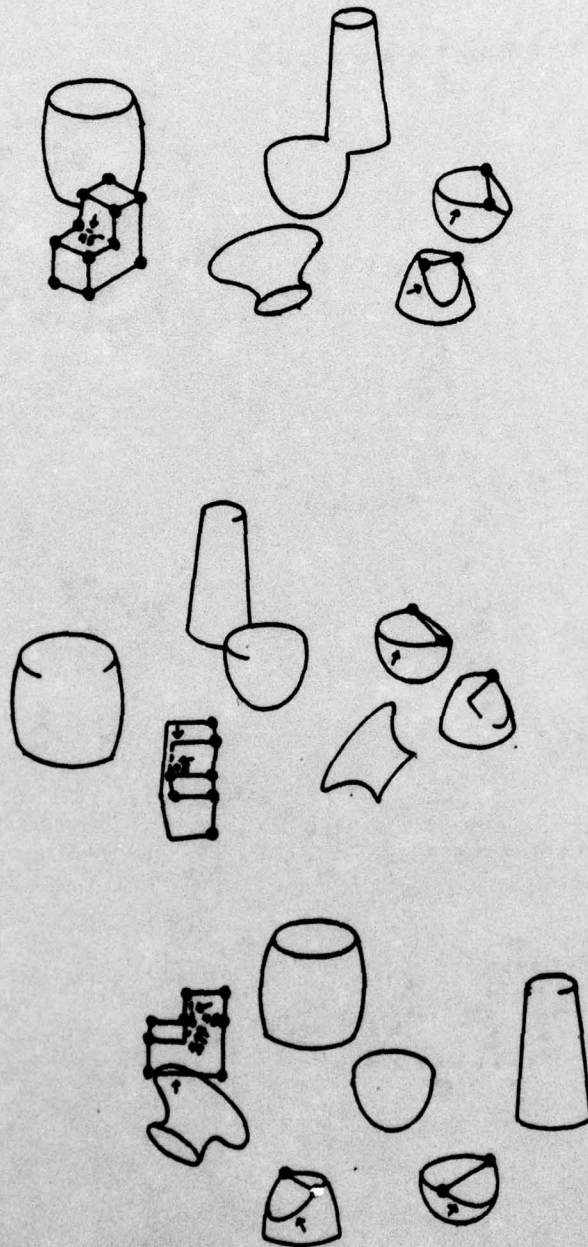


Fig. 6.6 Example 2 (For Explanation of Markings see Fig. 6.1)

A TRIPLE WAS FOUND	LINE	27	LINE	39	LINE
20					
A TRIPLE WAS FOUND	LINE	25	LINE	38	LINE
37					
A TRIPLE WAS FOUND	LINE	24	LINE	36	LINE
38					
A TRIPLE WAS FOUND	LINE	30	LINE	30	LINE
45					
A TRIPLE WAS FOUND	LINE	31	LINE	14	LINE
17					
A TRIPLE WAS FOUND	LINE	32	LINE	10	LINE
30					
-----					
PICTON ASSEMBLY					
-----					
MEMBERS OF FACE GROUP NUMBER		1			
PI6 PARAMETERS	34		29		
SINGLE MEMBER REGION	34		1		
TIMES OF FACE GROUP ARE	34		18		
	1				
	38		1		
	1				
IN PICTURE	3		IN PICTURE	39	IN PICTURE
PI6 PARAMETERS	31		16	3	15
SINGLE MEMBER REGION	31				
TIMES	31		31		
MEMBERS OF FACE GROUP NUMBER	31		1		
	2		17		

Fig. 6.7 (a) Computer Output Data for Example 2  
(Continued)



- 84 -

Fig. 6.7 (b) Computer Output Data for Example 2  
(Continued)

MEMBERS OF FACE GROUP NUMBER	28	47		
PIA PARAMETERS	27	11		
SINGLE MEMBER REGION	27	1		
LINES	27	19		
MEMBERS OF FACE GROUP	11			
ADF	44			
	2			
MEMBERS OF FACE GROUP NUMBER	2			
PIA PARAMETERS	27	12		
SINGLE MEMBER REGION	27	1		
LINES	27	39		
MEMBERS OF FACE GROUP NUMBER	27	13		
PIA PARAMETERS	25	1		
SINGLE MEMBER REGION	25	38		
LINES	25	38		
MEMBERS OF FACE GROUP	13			
ADF	42			
	2			
IN PICTURE	3	41		
PIA PARAMETERS	24	1		
SINGLE MEMBER REGION	24	36		
LINES	24	14		
MEMBERS OF FACE GROUP NUMBER	24			
PIA PARAMETERS	25	1		
SINGLE MEMBER REGION	25	38		
LINES	25	15		
MEMBERS OF FACE GROUP NUMBER	25			
PIA PARAMETERS	24	1		
SINGLE MEMBER REGION	24	36		
LINES	24	16		
MEMBERS OF FACE GROUP NUMBER	24			
PIA PARAMETERS	21	1		
LINES	21	48		
MEMBERS OF FACE GROUP NUMBER	17	2		
PIA PARAMETERS	19	17		
LINES	19	1		
MEMBERS OF FACE GROUP NUMBER	16	5		
PIA PARAMETERS	18	6		
LINES	18	18		
MEMBERS OF FACE GROUP NUMBER	16	1		
PIA PARAMETERS	15	7		
LINES	15	6		
MEMBERS OF FACE GROUP NUMBER	15	19		
PIA PARAMETERS	15	1		
LINES	15	NONE		
MEMBERS OF FACE GROUP NUMBER	3	19		
PIA PARAMETERS	15	20		
LINES	15			
MEMBERS OF FACE GROUP NUMBER	15	1		
PIA PARAMETERS	15	NONE		
LINES	15			

MEMBERS OF FACE GROUP NUMBER	28	47		
PIA PARAMETERS	27	11		
SINGLE MEMBER REGION	27	1		
LINES	27	19		
MEMBERS OF FACE GROUP	11			
ADF	44			
	2			
MEMBERS OF FACE GROUP NUMBER	2			
PIA PARAMETERS	27	12		
SINGLE MEMBER REGION	27	1		
LINES	27	39		
MEMBERS OF FACE GROUP NUMBER	27	13		
PIA PARAMETERS	25	1		
SINGLE MEMBER REGION	25	38		
LINES	25	38		
MEMBERS OF FACE GROUP	13			
ADF	42			
	2			
IN PICTURE	3	41		
PIA PARAMETERS	24	1		
SINGLE MEMBER REGION	24	36		
LINES	24	14		
MEMBERS OF FACE GROUP NUMBER	24			
PIA PARAMETERS	25	1		
SINGLE MEMBER REGION	25	38		
LINES	25	15		
MEMBERS OF FACE GROUP NUMBER	25			
PIA PARAMETERS	24	1		
SINGLE MEMBER REGION	24	36		
LINES	24	16		
MEMBERS OF FACE GROUP NUMBER	24			
PIA PARAMETERS	21	1		
LINES	21	48		
MEMBERS OF FACE GROUP NUMBER	17	2		
PIA PARAMETERS	19	17		
LINES	19	1		
MEMBERS OF FACE GROUP NUMBER	16	5		
PIA PARAMETERS	18	6		
LINES	18	18		
MEMBERS OF FACE GROUP NUMBER	16	1		
PIA PARAMETERS	15	7		
LINES	15	6		
MEMBERS OF FACE GROUP NUMBER	15	19		
PIA PARAMETERS	15	1		
LINES	15	NONE		
MEMBERS OF FACE GROUP NUMBER	3	19		
PIA PARAMETERS	15	20		
LINES	15			
MEMBERS OF FACE GROUP NUMBER	15	1		
PIA PARAMETERS	15	NONE		
LINES	15			

Fig. 6.7 (c) Computer Output Data for Example 2  
(Continued)



AD-A035 563

RENSSELAER POLYTECHNIC INST TROY N Y COMPUTER RESEAR--ETC F/G 9/2  
COMPUTER RECONSTRUCTION OF BODIES BOUNDED BY QUADRIC SURFACES F--ETC(11)  
SEP 76 R SHAPIRA

AF-AFOSR-2937-76

UNCLASSIFIED

CRL-48

AFOSR-TR-77-0051

NL

2 OF 2

AD  
A035563



END

DATE  
FILMED

3-77

```

LINE 6 20
LINE 5 21
LINE 10 22
LINE 11 23
LINE 14 NONE
MEMBERS OF FACE GROUP NUMBER 21
*****
D16 PARAMETERS 14 1
LINE 14 12
LINE 13 15
MEMBERS OF FACE GROUP NUMBER 22
*****
D16 PARAMETERS 13 1
LINE 13 15
LINE 11 23
LINE 9 16
LINE 12 16
MEMBERS OF FACE GROUP NUMBER 23
*****
D16 PARAMETERS 10 1
LINE 10 22
LINE 45 17
LINE 8 25
LINE 9 16
LINE 16 24
MEMBERS OF FACE GROUP NUMBER 25
*****
D16 PARAMETERS 7 1
LINE 7 48
LINE 45 17
LINE 5 21
LINE 1 18
MEMBERS OF FACE GROUP NUMBER 26
*****
D16 PARAMETERS 4 1
LINE 4 20
LINE 3 19
LINE 2 18
LINE 1 18
*****
OBJECTS FORMATIONS
*****
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER 1
*****
FACE GROUP 1 1 FACE GROUP
FACE GROUP 1 1 AND FACE GROUP
FACE GROUP 1 1 AND FACE GROUP
FACE GROUP 1 1 HAS A LIMB AND MUST BE CURVED
*****
THE FOLLOWING FACES ARE CURVED:
*****
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER 2
*****

```

Fig. 6.7 (d) Computer Output Data for Example 2 (Continued)



FACE GROUP	3	FACE GROUP	4	16
FACE GROUP	3	AND FACE GROUP	4	EDGE
FACE GROUP	3	AND FACE GROUP	16	EDGE
FACE GROUP	3	HAS A LINK AND MUST BE CURVED		
THE FOLLOWING FACES ARE CURVED:				
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER 3				
FACE GROUP	5	FACE GROUP	6	17
FACE GROUP	10	AND FACE GROUP	6	EDGE
FACE GROUP	5	AND FACE GROUP	17	EDGE
FACE GROUP	5	AND FACE GROUP	18	EDGE
FACE GROUP	5	HAS A LINK AND MUST BE CURVED		
THE FOLLOWING FACES ARE CURVED:				
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER 4				
FACE GROUP	8	FACE GROUP	9	10
FACE GROUP	8	AND FACE GROUP	9	EDGE
FACE GROUP	8	AND FACE GROUP	13	EDGE
FACE GROUP	8	HAS A LINK AND MUST BE CURVED		
THE FOLLOWING FACES ARE CURVED:				
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER 5				
FACE GROUP	11	FACE GROUP	12	
FACE GROUP	11	AND FACE GROUP	12	EDGE
FACE GROUP	11	HAS A LINK AND MUST BE CURVED		
THE FOLLOWING FACES ARE CURVED:				
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER 6				
FACE GROUP	13	FACE GROUP	14	15
FACE GROUP	13	AND FACE GROUP	14	EDGE
FACE GROUP	13	AND FACE GROUP	15	EDGE
FACE GROUP	13	HAS A LINK AND MUST BE CURVED		
THE FOLLOWING FACES ARE CURVED:				
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER 7				
FACE GROUP	19	FACE GROUP	20	21
FACE GROUP	22	FACE GROUP	23	24
FACE GROUP	25	FACE GROUP	26	
FACE GROUP	19	AND FACE GROUP	20	EDGE
FACE GROUP	25	AND FACE GROUP	26	EDGE
THE FOLLOWING FACES ARE CURVED:				
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER 8				
FACE GROUP	25	FACE GROUP	25	25

Fig. 6.7 (e) Computer Output Data for Example 2  
(Continued)

# ACCOUNTING LOG

## STEP ACCOUNTING

PROGRAM	CPU SEC.	ERT SEC.	TPE I/O	DSK I/O	GET I/O	REGION K	LDRE K	CARDS K	SP	LINES SP	M. UNITS
IFMAA	9.48	12.41		90		300	304			1.351	135
IFMIF-800	2.00	20.98		981		128	124			26	228
PCMAO-110	3.91	4.43		1		360	336			1.406	73
TOTAL	15.39	45.82		1.072						2.783	497

UNIT RECORD : BYTES TRANS. CARDS RD PRI CCMS  
 685 2.960

JOB COST (EXCLUDES PAPER AND PUNCH) \$6.57

PAGES PRINTED M. UNITS PAPER COST  
 64 441 4.411

TOTAL M. UNITS 875

DATE=75.714 TIMES= MON-FRI: 12.46.58 OFF: 12.47.33 EXFC-ON: 16.40.09 OFF: 16.47.18 PRI-JN: 16.51.66 OFF: 16.54.47

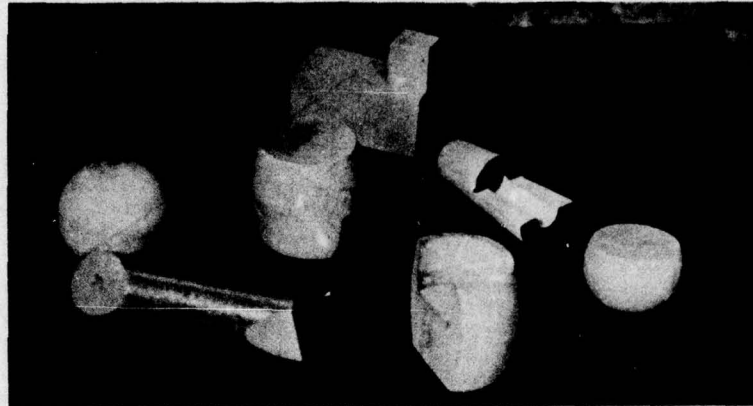
JOB	CPU SEC.	ERT SEC.	TPE I/O	DSK I/O	GET I/O	CARDS RD	CARDS PJ	LINES PR	PAGES PJ	K-BYTE\$-SEC
14	279	58%		12.743		8.427		20.712	629	115.119

ACCOUNT USAGE TO DATE \$100.70 : SHY20 - BUDGET BALANCE \$ 358.33 22.2 %

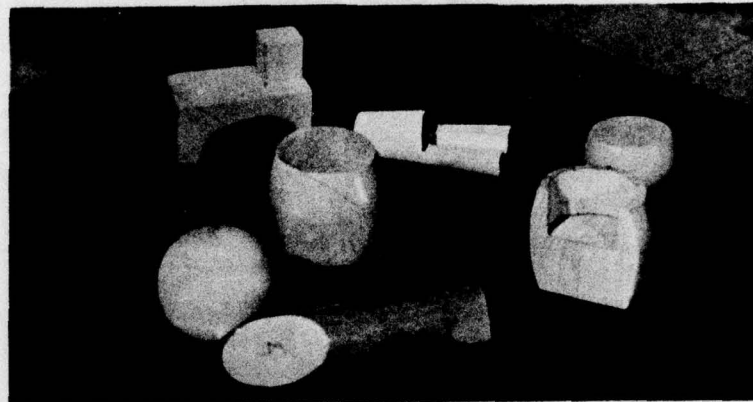
ORDER USAGE TO DATE 1.009.1711 : CMT70 - ORDER RANGE 469.5711 71.2 %

Fig. 6.7 (f) Computer Output Data for Example 2 (End)





(a)



(b)



(c)

Fig. 6.8 Third Scene Example. (a) Picture 1,  
(b) Picture 2, (c) Picture 3

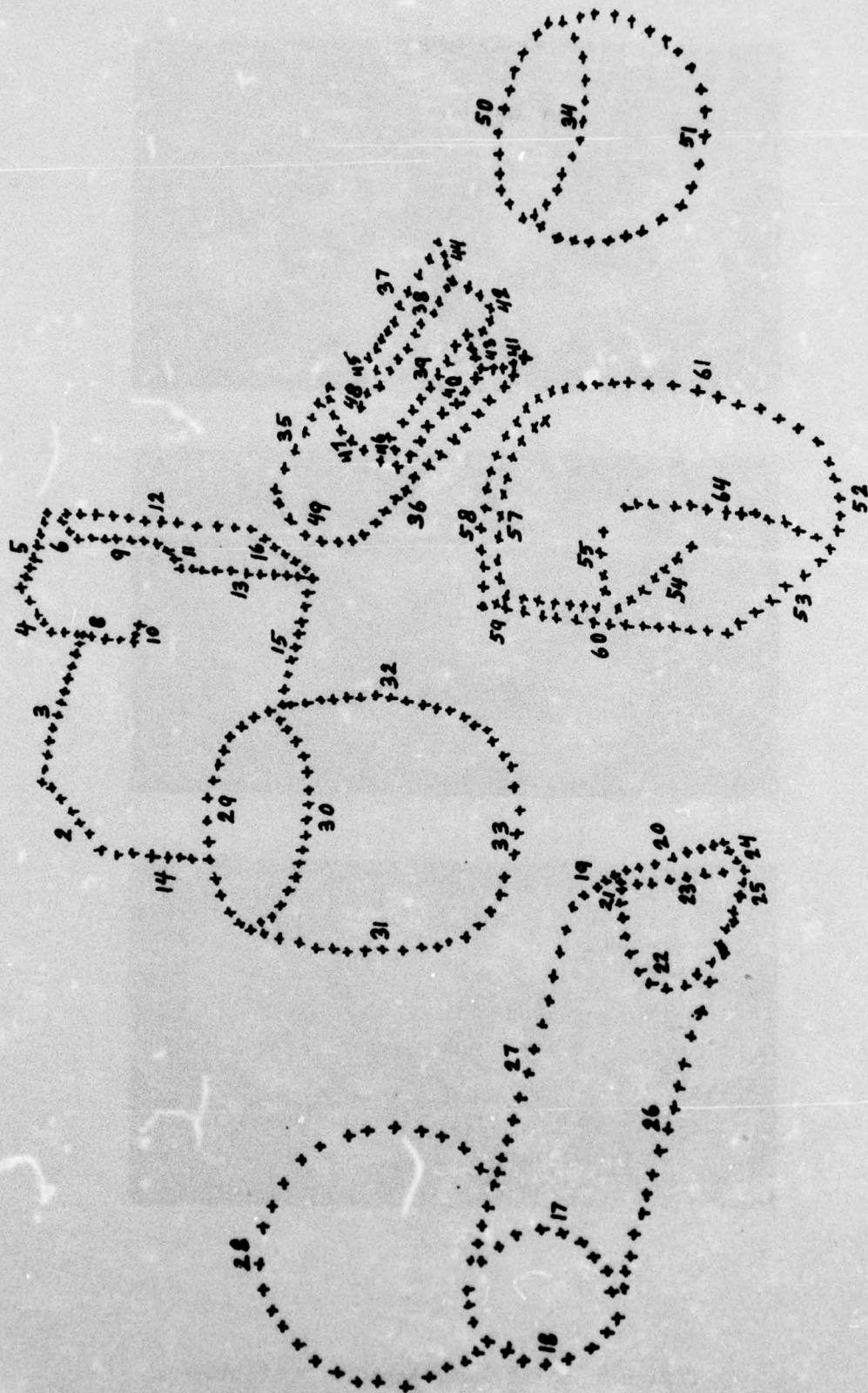


Fig. 6.9 Third Scene: Picture-1 Input Data



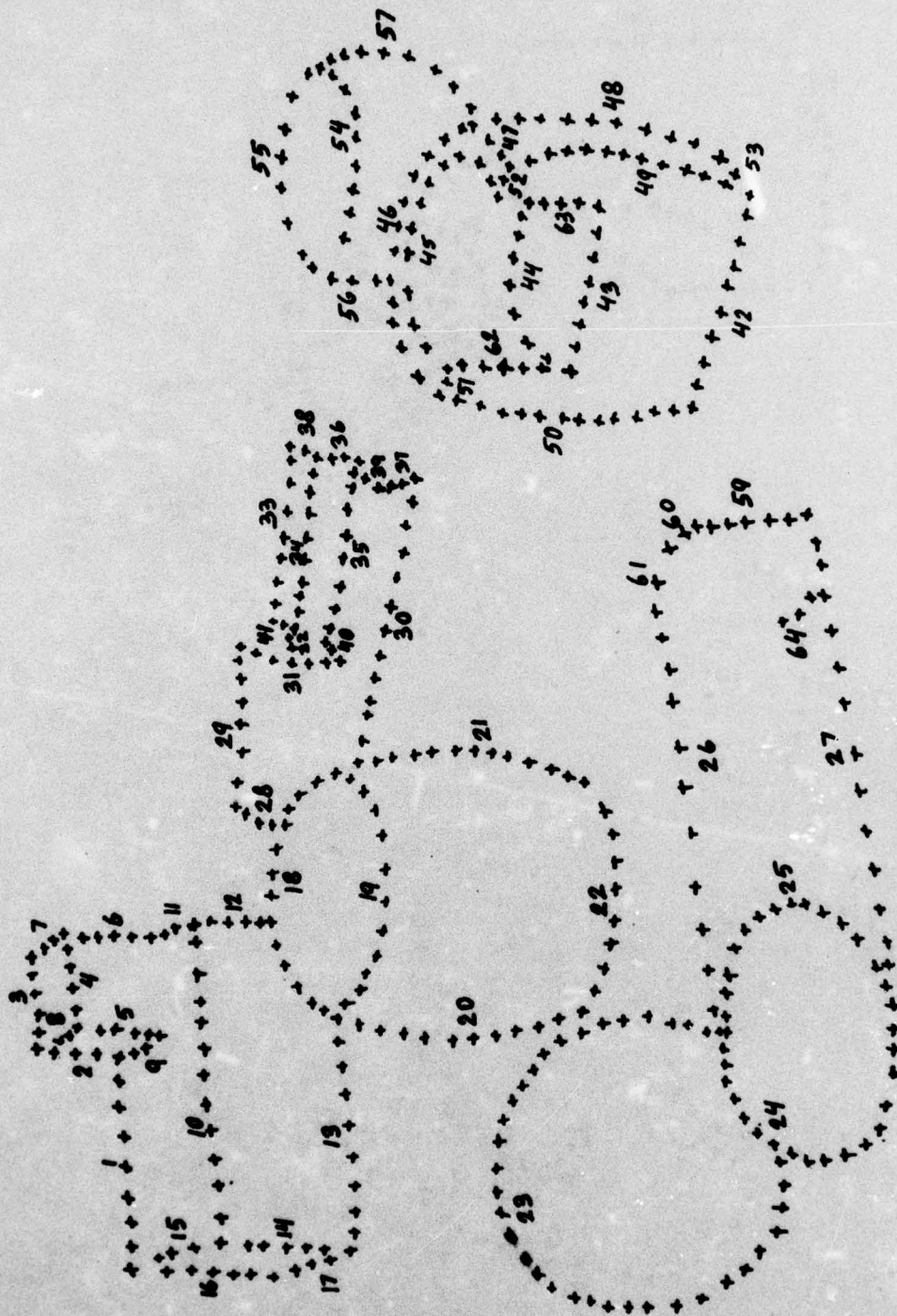


Fig. 6.10 Third Scene: Picture-2 Input Data

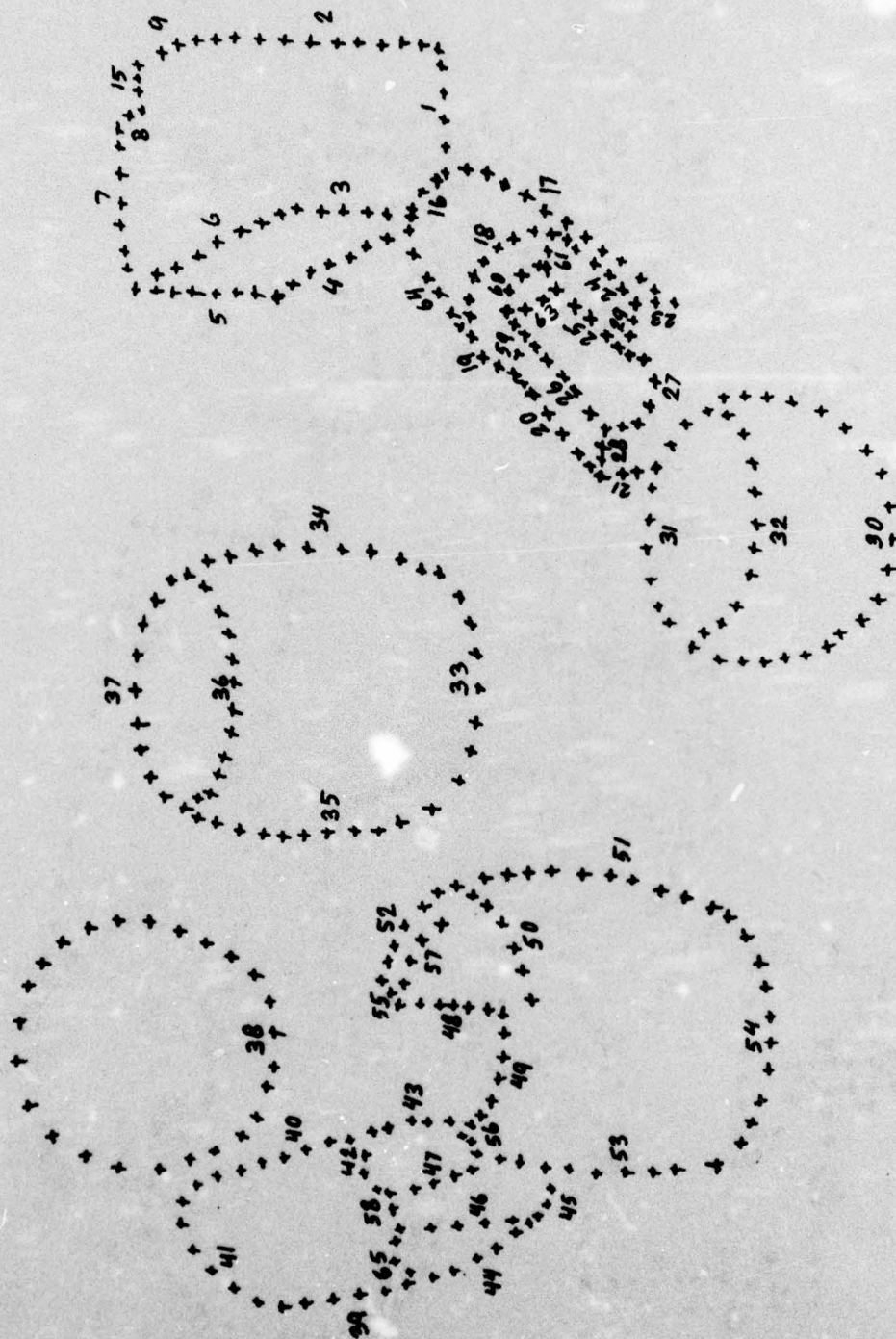
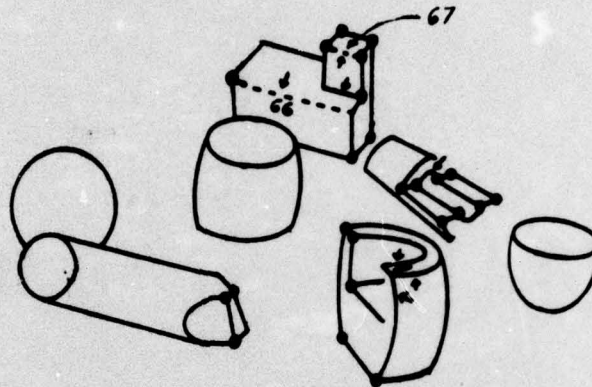


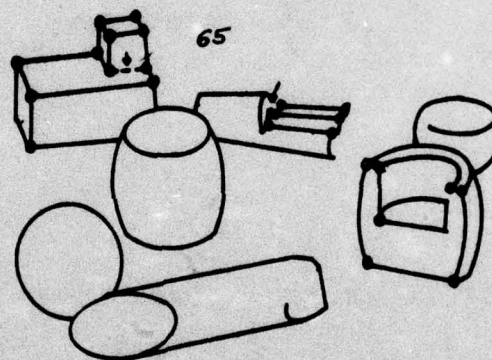
Fig. 6.11 Third Scene: Picture-3 Input Data



PICTURE 1



PICTURE 2



PICTURE 3

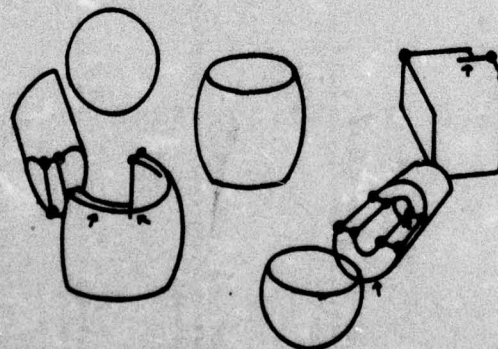


Fig. 6.12 Example 3 (For Explanation of Markings  
see Figure 6.1)





Fig. 6.13 (b) Computer Output Data for Example 3  
(Continued)

SINGLE MEMBER REGION	23	IN PICTURE	1	IN PICTURE	1
LINE	22		22		
LINE	14				
LINE	34				
IN PICTURE	1	IN PICTURE	1	IN PICTURE	21
IN PICTURE	2				2
IN PICTURE	3				3
IN PICTURE	24		24		
IN PICTURE	25		1		
IN PICTURE	26		20		
IN PICTURE	27		34		
IN PICTURE	28				
IN PICTURE	29				
IN PICTURE	30				
IN PICTURE	31				
IN PICTURE	32				
IN PICTURE	33				
IN PICTURE	34				
IN PICTURE	35				
IN PICTURE	36				
IN PICTURE	37				
IN PICTURE	38				
IN PICTURE	39				
IN PICTURE	40				
IN PICTURE	41				
IN PICTURE	42				
IN PICTURE	43				
IN PICTURE	44				
IN PICTURE	45				
IN PICTURE	46				
IN PICTURE	47				
IN PICTURE	48				
IN PICTURE	49				
IN PICTURE	50				
IN PICTURE	51				
IN PICTURE	52				
IN PICTURE	53				
IN PICTURE	54				
IN PICTURE	55				
IN PICTURE	56				
IN PICTURE	57				
IN PICTURE	58				
IN PICTURE	59				
IN PICTURE	60				
IN PICTURE	61				
IN PICTURE	62				
IN PICTURE	63				
IN PICTURE	64				
IN PICTURE	65				
IN PICTURE	66				
IN PICTURE	67				
IN PICTURE	68				
IN PICTURE	69				
IN PICTURE	70				
IN PICTURE	71				
IN PICTURE	72				
IN PICTURE	73				
IN PICTURE	74				
IN PICTURE	75				
IN PICTURE	76				
IN PICTURE	77				
IN PICTURE	78				
IN PICTURE	79				
IN PICTURE	80				
IN PICTURE	81				
IN PICTURE	82				
IN PICTURE	83				
IN PICTURE	84				
IN PICTURE	85				
IN PICTURE	86				
IN PICTURE	87				
IN PICTURE	88				
IN PICTURE	89				
IN PICTURE	90				
IN PICTURE	91				
IN PICTURE	92				
IN PICTURE	93				
IN PICTURE	94				
IN PICTURE	95				
IN PICTURE	96				
IN PICTURE	97				
IN PICTURE	98				
IN PICTURE	99				
IN PICTURE	100				

Fig. 6.13 (c) Computer Output Data for Example 3  
(Continued)



- 97 -

Computer Out  
(Continued)

LINE	NUMBER OF FACE GROUP	NAME	FACE	NAME
1	1	FACE GROUP	1	FACE GROUP
2	2	FACE GROUP	2	FACE GROUP
3	3	FACE GROUP	3	FACE GROUP
4	4	FACE GROUP	4	FACE GROUP
5	5	FACE GROUP	5	FACE GROUP
6	6	FACE GROUP	6	FACE GROUP
7	7	FACE GROUP	7	FACE GROUP
8	8	FACE GROUP	8	FACE GROUP
9	9	FACE GROUP	9	FACE GROUP
10	10	FACE GROUP	10	FACE GROUP
11	11	FACE GROUP	11	FACE GROUP
12	12	FACE GROUP	12	FACE GROUP
13	13	FACE GROUP	13	FACE GROUP
14	14	FACE GROUP	14	FACE GROUP
15	15	FACE GROUP	15	FACE GROUP
16	16	FACE GROUP	16	FACE GROUP
17	17	FACE GROUP	17	FACE GROUP
18	18	FACE GROUP	18	FACE GROUP
19	19	FACE GROUP	19	FACE GROUP
20	20	FACE GROUP	20	FACE GROUP
21	21	FACE GROUP	21	FACE GROUP
22	22	FACE GROUP	22	FACE GROUP
23	23	FACE GROUP	23	FACE GROUP
24	24	FACE GROUP	24	FACE GROUP
25	25	FACE GROUP	25	FACE GROUP
26	26	FACE GROUP	26	FACE GROUP
27	27	FACE GROUP	27	FACE GROUP
28	28	FACE GROUP	28	FACE GROUP
29	29	FACE GROUP	29	FACE GROUP
30	30	FACE GROUP	30	FACE GROUP
31	31	FACE GROUP	31	FACE GROUP
32	32	FACE GROUP	32	FACE GROUP
33	33	FACE GROUP	33	FACE GROUP
34	34	FACE GROUP	34	FACE GROUP
35	35	FACE GROUP	35	FACE GROUP
36	36	FACE GROUP	36	FACE GROUP
37	37	FACE GROUP	37	FACE GROUP
38	38	FACE GROUP	38	FACE GROUP
39	39	FACE GROUP	39	FACE GROUP
40	40	FACE GROUP	40	FACE GROUP
41	41	FACE GROUP	41	FACE GROUP
42	42	FACE GROUP	42	FACE GROUP
43	43	FACE GROUP	43	FACE GROUP
44	44	FACE GROUP	44	FACE GROUP
45	45	FACE GROUP	45	FACE GROUP
46	46	FACE GROUP	46	FACE GROUP
47	47	FACE GROUP	47	FACE GROUP
48	48	FACE GROUP	48	FACE GROUP
49	49	FACE GROUP	49	FACE GROUP
50	50	FACE GROUP	50	FACE GROUP
51	51	FACE GROUP	51	FACE GROUP
52	52	FACE GROUP	52	FACE GROUP
53	53	FACE GROUP	53	FACE GROUP
54	54	FACE GROUP	54	FACE GROUP
55	55	FACE GROUP	55	FACE GROUP
56	56	FACE GROUP	56	FACE GROUP
57	57	FACE GROUP	57	FACE GROUP
58	58	FACE GROUP	58	FACE GROUP
59	59	FACE GROUP	59	FACE GROUP
60	60	FACE GROUP	60	FACE GROUP
61	61	FACE GROUP	61	FACE GROUP
62	62	FACE GROUP	62	FACE GROUP
63	63	FACE GROUP	63	FACE GROUP
64	64	FACE GROUP	64	FACE GROUP
65	65	FACE GROUP	65	FACE GROUP
66	66	FACE GROUP	66	FACE GROUP
67	67	FACE GROUP	67	FACE GROUP
68	68	FACE GROUP	68	FACE GROUP
69	69	FACE GROUP	69	FACE GROUP
70	70	FACE GROUP	70	FACE GROUP
71	71	FACE GROUP	71	FACE GROUP
72	72	FACE GROUP	72	FACE GROUP
73	73	FACE GROUP	73	FACE GROUP
74	74	FACE GROUP	74	FACE GROUP
75	75	FACE GROUP	75	FACE GROUP
76	76	FACE GROUP	76	FACE GROUP
77	77	FACE GROUP	77	FACE GROUP
78	78	FACE GROUP	78	FACE GROUP
79	79	FACE GROUP	79	FACE GROUP
80	80	FACE GROUP	80	FACE GROUP
81	81	FACE GROUP	81	FACE GROUP
82	82	FACE GROUP	82	FACE GROUP
83	83	FACE GROUP	83	FACE GROUP
84	84	FACE GROUP	84	FACE GROUP
85	85	FACE GROUP	85	FACE GROUP
86	86	FACE GROUP	86	FACE GROUP
87	87	FACE GROUP	87	FACE GROUP
88	88	FACE GROUP	88	FACE GROUP
89	89	FACE GROUP	89	FACE GROUP
90	90	FACE GROUP	90	FACE GROUP
91	91	FACE GROUP	91	FACE GROUP
92	92	FACE GROUP	92	FACE GROUP
93	93	FACE GROUP	93	FACE GROUP
94	94	FACE GROUP	94	FACE GROUP
95	95	FACE GROUP	95	FACE GROUP
96	96	FACE GROUP	96	FACE GROUP
97	97	FACE GROUP	97	FACE GROUP
98	98	FACE GROUP	98	FACE GROUP
99	99	FACE GROUP	99	FACE GROUP
100	100	FACE GROUP	100	FACE GROUP

Fig. 6.13 (e) Computer Output Data for Example 3  
(Continued)



```

THE FOLLOWING FACE GROUPS FROM UNITS SUMMEN
FACE GROUP 21 FACE GROUP 22 23
FACE GROUP 24 25 26
FACE GROUP 27 28 29
FACE GROUP 30 31 32
FACE GROUP 33 34 35
FACE GROUP 36 37 38
FACE GROUP 39 40 41
FACE GROUP 42 43 44
FACE GROUP 45 46 47
FACE GROUP 48 49 50
FACE GROUP 51 52 53
FACE GROUP 54 55 56
FACE GROUP 57 58 59
FACE GROUP 60 61 62
FACE GROUP 63 64 65
FACE GROUP 66 67 68
FACE GROUP 69 70 71
FACE GROUP 72 73 74
FACE GROUP 75 76 77
FACE GROUP 78 79 80
FACE GROUP 81 82 83
FACE GROUP 84 85 86
FACE GROUP 87 88 89
FACE GROUP 90 91 92
FACE GROUP 93 94 95
FACE GROUP 96 97 98
FACE GROUP 99 100 101
FACE GROUP 102 103 104
FACE GROUP 105 106 107
FACE GROUP 108 109 110
FACE GROUP 111 112 113
FACE GROUP 114 115 116
FACE GROUP 117 118 119
FACE GROUP 120 121 122
FACE GROUP 123 124 125
FACE GROUP 126 127 128
FACE GROUP 129 130 131
FACE GROUP 132 133 134
FACE GROUP 135 136 137
FACE GROUP 138 139 140
FACE GROUP 141 142 143
FACE GROUP 144 145 146
FACE GROUP 147 148 149
FACE GROUP 150 151 152
FACE GROUP 153 154 155
FACE GROUP 156 157 158
FACE GROUP 159 160 161
FACE GROUP 162 163 164
FACE GROUP 165 166 167
FACE GROUP 168 169 170
FACE GROUP 171 172 173
FACE GROUP 174 175 176
FACE GROUP 177 178 179
FACE GROUP 180 181 182
FACE GROUP 183 184 185
FACE GROUP 186 187 188
FACE GROUP 189 190 191
FACE GROUP 192 193 194
FACE GROUP 195 196 197
FACE GROUP 198 199 200
FACE GROUP 201 202 203
FACE GROUP 204 205 206
FACE GROUP 207 208 209
FACE GROUP 210 211 212
FACE GROUP 213 214 215
FACE GROUP 216 217 218
FACE GROUP 219 220 221
FACE GROUP 222 223 224
FACE GROUP 225 226 227
FACE GROUP 228 229 230
FACE GROUP 231 232 233
FACE GROUP 234 235 236
FACE GROUP 237 238 239
FACE GROUP 240 241 242
FACE GROUP 243 244 245
FACE GROUP 246 247 248
FACE GROUP 249 250 251
FACE GROUP 252 253 254
FACE GROUP 255 256 257
FACE GROUP 258 259 260
FACE GROUP 261 262 263
FACE GROUP 264 265 266
FACE GROUP 267 268 269
FACE GROUP 270 271 272
FACE GROUP 273 274 275
FACE GROUP 276 277 278
FACE GROUP 279 280 281
FACE GROUP 282 283 284
FACE GROUP 285 286 287
FACE GROUP 288 289 290
FACE GROUP 291 292 293
FACE GROUP 294 295 296
FACE GROUP 297 298 299
FACE GROUP 300 301 302
FACE GROUP 303 304 305
FACE GROUP 306 307 308
FACE GROUP 309 310 311
FACE GROUP 312 313 314
FACE GROUP 315 316 317
FACE GROUP 318 319 320
FACE GROUP 321 322 323
FACE GROUP 324 325 326
FACE GROUP 327 328 329
FACE GROUP 330 331 332
FACE GROUP 333 334 335
FACE GROUP 336 337 338
FACE GROUP 339 340 341
FACE GROUP 342 343 344
FACE GROUP 345 346 347
FACE GROUP 348 349 350
FACE GROUP 351 352 353
FACE GROUP 354 355 356
FACE GROUP 357 358 359
FACE GROUP 360 361 362
FACE GROUP 363 364 365
FACE GROUP 366 367 368
FACE GROUP 369 370 371
FACE GROUP 372 373 374
FACE GROUP 375 376 377
FACE GROUP 378 379 380
FACE GROUP 381 382 383
FACE GROUP 384 385 386
FACE GROUP 387 388 389
FACE GROUP 390 391 392
FACE GROUP 393 394 395
FACE GROUP 396 397 398
FACE GROUP 399 400 401
FACE GROUP 402 403 404
FACE GROUP 405 406 407
FACE GROUP 408 409 410
FACE GROUP 411 412 413
FACE GROUP 414 415 416
FACE GROUP 417 418 419
FACE GROUP 420 421 422
FACE GROUP 423 424 425
FACE GROUP 426 427 428
FACE GROUP 429 430 431
FACE GROUP 432 433 434
FACE GROUP 435 436 437
FACE GROUP 438 439 440
FACE GROUP 441 442 443
FACE GROUP 444 445 446
FACE GROUP 447 448 449
FACE GROUP 450 451 452
FACE GROUP 453 454 455
FACE GROUP 456 457 458
FACE GROUP 459 460 461
FACE GROUP 462 463 464
FACE GROUP 465 466 467
FACE GROUP 468 469 470
FACE GROUP 471 472 473
FACE GROUP 474 475 476
FACE GROUP 477 478 479
FACE GROUP 480 481 482
FACE GROUP 483 484 485
FACE GROUP 486 487 488
FACE GROUP 489 490 491
FACE GROUP 492 493 494
FACE GROUP 495 496 497
FACE GROUP 498 499 500
FACE GROUP 501 502 503
FACE GROUP 504 505 506
FACE GROUP 507 508 509
FACE GROUP 510 511 512
FACE GROUP 513 514 515
FACE GROUP 516 517 518
FACE GROUP 519 520 521
FACE GROUP 522 523 524
FACE GROUP 525 526 527
FACE GROUP 528 529 530
FACE GROUP 531 532 533
FACE GROUP 534 535 536
FACE GROUP 537 538 539
FACE GROUP 540 541 542
FACE GROUP 543 544 545
FACE GROUP 546 547 548
FACE GROUP 549 550 551
FACE GROUP 552 553 554
FACE GROUP 555 556 557
FACE GROUP 558 559 560
FACE GROUP 561 562 563
FACE GROUP 564 565 566
FACE GROUP 567 568 569
FACE GROUP 570 571 572
FACE GROUP 573 574 575
FACE GROUP 576 577 578
FACE GROUP 579 580 581
FACE GROUP 582 583 584
FACE GROUP 585 586 587
FACE GROUP 588 589 590
FACE GROUP 591 592 593
FACE GROUP 594 595 596
FACE GROUP 597 598 599
FACE GROUP 600 601 602
FACE GROUP 603 604 605
FACE GROUP 606 607 608
FACE GROUP 609 610 611
FACE GROUP 612 613 614
FACE GROUP 615 616 617
FACE GROUP 618 619 620
FACE GROUP 621 622 623
FACE GROUP 624 625 626
FACE GROUP 627 628 629
FACE GROUP 630 631 632
FACE GROUP 633 634 635
FACE GROUP 636 637 638
FACE GROUP 639 640 641
FACE GROUP 642 643 644
FACE GROUP 645 646 647
FACE GROUP 648 649 650
FACE GROUP 651 652 653
FACE GROUP 654 655 656
FACE GROUP 657 658 659
FACE GROUP 660 661 662
FACE GROUP 663 664 665
FACE GROUP 666 667 668
FACE GROUP 669 670 671
FACE GROUP 672 673 674
FACE GROUP 675 676 677
FACE GROUP 678 679 680
FACE GROUP 681 682 683
FACE GROUP 684 685 686
FACE GROUP 687 688 689
FACE GROUP 690 691 692
FACE GROUP 693 694 695
FACE GROUP 696 697 698
FACE GROUP 699 700 701
FACE GROUP 702 703 704
FACE GROUP 705 706 707
FACE GROUP 708 709 710
FACE GROUP 711 712 713
FACE GROUP 714 715 716
FACE GROUP 717 718 719
FACE GROUP 720 721 722
FACE GROUP 723 724 725
FACE GROUP 726 727 728
FACE GROUP 729 730 731
FACE GROUP 732 733 734
FACE GROUP 735 736 737
FACE GROUP 738 739 740
FACE GROUP 741 742 743
FACE GROUP 744 745 746
FACE GROUP 747 748 749
FACE GROUP 750 751 752
FACE GROUP 753 754 755
FACE GROUP 756 757 758
FACE GROUP 759 760 761
FACE GROUP 762 763 764
FACE GROUP 765 766 767
FACE GROUP 768 769 770
FACE GROUP 771 772 773
FACE GROUP 774 775 776
FACE GROUP 777 778 779
FACE GROUP 780 781 782
FACE GROUP 783 784 785
FACE GROUP 786 787 788
FACE GROUP 789 790 791
FACE GROUP 792 793 794
FACE GROUP 795 796 797
FACE GROUP 798 799 800
FACE GROUP 801 802 803
FACE GROUP 804 805 806
FACE GROUP 807 808 809
FACE GROUP 810 811 812
FACE GROUP 813 814 815
FACE GROUP 816 817 818
FACE GROUP 819 820 821
FACE GROUP 822 823 824
FACE GROUP 825 826 827
FACE GROUP 828 829 830
FACE GROUP 831 832 833
FACE GROUP 834 835 836
FACE GROUP 837 838 839
FACE GROUP 840 841 842
FACE GROUP 843 844 845
FACE GROUP 846 847 848
FACE GROUP 849 850 851
FACE GROUP 852 853 854
FACE GROUP 855 856 857
FACE GROUP 858 859 860
FACE GROUP 861 862 863
FACE GROUP 864 865 866
FACE GROUP 867 868 869
FACE GROUP 870 871 872
FACE GROUP 873 874 875
FACE GROUP 876 877 878
FACE GROUP 879 880 881
FACE GROUP 882 883 884
FACE GROUP 885 886 887
FACE GROUP 888 889 890
FACE GROUP 891 892 893
FACE GROUP 894 895 896
FACE GROUP 897 898 899
FACE GROUP 900 901 902
FACE GROUP 903 904 905
FACE GROUP 906 907 908
FACE GROUP 909 910 911
FACE GROUP 912 913 914
FACE GROUP 915 916 917
FACE GROUP 918 919 920
FACE GROUP 921 922 923
FACE GROUP 924 925 926
FACE GROUP 927 928 929
FACE GROUP 930 931 932
FACE GROUP 933 934 935
FACE GROUP 936 937 938
FACE GROUP 939 940 941
FACE GROUP 942 943 944
FACE GROUP 945 946 947
FACE GROUP 948 949 950
FACE GROUP 951 952 953
FACE GROUP 954 955 956
FACE GROUP 957 958 959
FACE GROUP 960 961 962
FACE GROUP 963 964 965
FACE GROUP 966 967 968
FACE GROUP 969 970 971
FACE GROUP 972 973 974
FACE GROUP 975 976 977
FACE GROUP 978 979 980
FACE GROUP 981 982 983
FACE GROUP 984 985 986
FACE GROUP 987 988 989
FACE GROUP 990 991 992
FACE GROUP 993 994 995
FACE GROUP 996 997 998
FACE GROUP 999 1000 1001
FACE GROUP 1002 1003 1004
FACE GROUP 1005 1006 1007
FACE GROUP 1008 1009 1010
FACE GROUP 1011 1012 1013
FACE GROUP 1014 1015 1016
FACE GROUP 1017 1018 1019
FACE GROUP 1020 1021 1022
FACE GROUP 1023 1024 1025
FACE GROUP 1026 1027 1028
FACE GROUP 1029 1030 1031
FACE GROUP 1032 1033 1034
FACE GROUP 1035 1036 1037
FACE GROUP 1038 1039 1040
FACE GROUP 1041 1042 1043
FACE GROUP 1044 1045 1046
FACE GROUP 1047 1048 1049
FACE GROUP 1050 1051 1052
FACE GROUP 1053 1054 1055
FACE GROUP 1056 1057 1058
FACE GROUP 1059 1060 1061
FACE GROUP 1062 1063 1064
FACE GROUP 1065 1066 1067
FACE GROUP 1068 1069 1070
FACE GROUP 1071 1072 1073
FACE GROUP 1074 1075 1076
FACE GROUP 1077 1078 1079
FACE GROUP 1080 1081 1082
FACE GROUP 1083 1084 1085
FACE GROUP 1086 1087 1088
FACE GROUP 1089 1090 1091
FACE GROUP 1092 1093 1094
FACE GROUP 1095 1096 1097
FACE GROUP 1098 1099 1100
FACE GROUP 1101 1102 1103
FACE GROUP 1104 1105 1106
FACE GROUP 1107 1108 1109
FACE GROUP 1110 1111 1112
FACE GROUP 1113 1114 1115
FACE GROUP 1116 1117 1118
FACE GROUP 1119 1120 1121
FACE GROUP 1122 1123 1124
FACE GROUP 1125 1126 1127
FACE GROUP 1128 1129 1130
FACE GROUP 1131 1132 1133
FACE GROUP 1134 1135 1136
FACE GROUP 1137 1138 1139
FACE GROUP 1140 1141 1142
FACE GROUP 1143 1144 1145
FACE GROUP 1146 1147 1148
FACE GROUP 1149 1150 1151
FACE GROUP 1152 1153 1154
FACE GROUP 1155 1156 1157
FACE GROUP 1158 1159 1160
FACE GROUP 1161 1162 1163
FACE GROUP 1164 1165 1166
FACE GROUP 1167 1168 1169
FACE GROUP 1170 1171 1172
FACE GROUP 1173 1174 1175
FACE GROUP 1176 1177 1178
FACE GROUP 1179 1180 1181
FACE GROUP 1182 1183 1184
FACE GROUP 1185 1186 1187
FACE GROUP 1188 1189 1190
FACE GROUP 1191 1192 1193
FACE GROUP 1194 1195 1196
FACE GROUP 1197 1198 1199
FACE GROUP 1200 1201 1202
FACE GROUP 1203 1204 1205
FACE GROUP 1206 1207 1208
FACE GROUP 1209 1210 1211
FACE GROUP 1212 1213 1214
FACE GROUP 1215 1216 1217
FACE GROUP 1218 1219 1220
FACE GROUP 1221 1222 1223
FACE GROUP 1224 1225 1226
FACE GROUP 1227 1228 1229
FACE GROUP 1230 1231 1232
FACE GROUP 1233 1234 1235
FACE GROUP 1236 1237 1238
FACE GROUP 1239 1240 1241
FACE GROUP 1242 1243 1244
FACE GROUP 1245 1246 1247
FACE GROUP 1248 1249 1250
FACE GROUP 1251 1252 1253
FACE GROUP 1254 1255 1256
FACE GROUP 1257 1258 1259
FACE GROUP 1260 1261 1262
FACE GROUP 1263 1264 1265
FACE GROUP 1266 1267 1268
FACE GROUP 1269 1270 1271
FACE GROUP 1272 1273 1274
FACE GROUP 1275 1276 1277
FACE GROUP 1278 1279 1280
FACE GROUP 1281 1282 1283
FACE GROUP 1284 1285 1286
FACE GROUP 1287 1288 1289
FACE GROUP 1290 1291 1292
FACE GROUP 1293 1294 1295
FACE GROUP 1296 1297 1298
FACE GROUP 1299 1300 1301
FACE GROUP 1302 1303 1304
FACE GROUP 1305 1306 1307
FACE GROUP 1308 1309 1310
FACE GROUP 1311 1312 1313
FACE GROUP 1314 1315 1316
FACE GROUP 1317 1318 1319
FACE GROUP 1320 1321 1322
FACE GROUP 1323 1324 1325
FACE GROUP 1326 1327 1328
FACE GROUP 1329 1330 1331
FACE GROUP 1332 1333 1334
FACE GROUP 1335 1336 1337
FACE GROUP 1338 1339 1340
FACE GROUP 1341 1342 1343
FACE GROUP 1344 1345 1346
FACE GROUP 1347 1348 1349
FACE GROUP 1350 1351 1352
FACE GROUP 1353 1354 1355
FACE GROUP 1356 1357 1358
FACE GROUP 1359 1360 1361
FACE GROUP 1362 1363 1364
FACE GROUP 1365 1366 1367
FACE GROUP 1368 1369 1370
FACE GROUP 1371 1372 1373
FACE GROUP 1374 1375 1376
FACE GROUP 1377 1378 1379
FACE GROUP 1380 1381 1382
FACE GROUP 1383 1384 1385
FACE GROUP 1386 1387 1388
FACE GROUP 1389 1390 1391
FACE GROUP 1392 1393 1394
FACE GROUP 1395 1396 1397
FACE GROUP 1398 1399 1400
FACE GROUP 1401 1402 1403
FACE GROUP 1404 1405 1406
FACE GROUP 1407 1408 1409
FACE GROUP 1410 1411 1412
FACE GROUP 1413 1414 1415
FACE GROUP 1416 1417 1418
FACE GROUP 1419 1420 1421
FACE GROUP 1422 1423 1424
FACE GROUP 1425 1426 1427
FACE GROUP 1428 1429 1430
FACE GROUP 1431 1432 1433
FACE GROUP 1434 1435 1436
FACE GROUP 1437 1438 1439
FACE GROUP 1440 1441 1442
FACE GROUP 1443 1444 1445
FACE GROUP 1446 1447 1448
FACE GROUP 1449 1450 1451
FACE GROUP 1452 1453 1454
FACE GROUP 1455 1456 1457
FACE GROUP 1458 1459 1460
FACE GROUP 1461 1462 1463
FACE GROUP 1464 1465 1466
FACE GROUP 1467 1468 1469
FACE GROUP 1470 1471 1472
FACE GROUP 1473 1474 1475
FACE GROUP 1476 1477 1478
FACE GROUP 1479 1480 1481
FACE GROUP 1482 1483 1484
FACE GROUP 1485 1486 1487
FACE GROUP 1488 1489 1490
FACE GROUP 1491 1492 1493
FACE GROUP 1494 1495 1496
FACE GROUP 1497 1498 1499
FACE GROUP 1500 1501 1502
FACE GROUP 1503 1504 1505
FACE GROUP 1506 1507 1508
FACE GROUP 1509 1510 1511
FACE GROUP 1512 1513 1514
FACE GROUP 1515 1516 1517
FACE GROUP 1518 1519 1520
FACE GROUP 1521 1522 1523
FACE GROUP 1524 1525 1526
FACE GROUP 1527 1528 1529
FACE GROUP 1530 1531 1532
FACE GROUP 1533 1534 1535
FACE GROUP 1536 1537 1538
FACE GROUP 1539 1540 1541
FACE GROUP 1542 1543 1544
FACE GROUP 1545 1546 1547
FACE GROUP 1548 1549 1550
FACE GROUP 1551 1552 1553
FACE GROUP 1554 1555 1556
FACE GROUP 1557 1558 1559
FACE GROUP 1560 1561 1562
FACE GROUP 1563 1564 1565
FACE GROUP 1566 1567 1568
FACE GROUP 1569 1570 1571
FACE GROUP 1572 1573 1574
FACE GROUP 1575 1576 1577
FACE GROUP 1578 1579 1580
FACE GROUP 1581 1582 1583
FACE GROUP 1584 1585 1586
FACE GROUP 1587 1588 1589
FACE GROUP 1590 1591 1592
FACE GROUP 1593 1594 1595
FACE GROUP 1596 1597 1598
FACE GROUP 1599 1600 1601
FACE GROUP 1602 1603 1604
FACE GROUP 1605 1606 1607
FACE GROUP 1608 1609 1610
FACE GROUP 1611 1612 1613
FACE GROUP 1614 1615 1616
FACE GROUP 1617 1618 1619
FACE GROUP 1620 1621 1622
FACE GROUP 1623 1624 1625
FACE GROUP 1626 1627 1628
FACE GROUP 1629 1630 1631
FACE GROUP 1632 1633 1634
FACE GROUP 1635 1636 1637
FACE GROUP 1638 1639 1640
FACE GROUP 1641 1642 1643
FACE GROUP 1644 1645 1646
FACE GROUP 1647 1648 1649
FACE GROUP 1650 1651 1652
FACE GROUP 1653 1654 1655
FACE GROUP 1656 1657 1658
FACE GROUP 1659 1660 1661
FACE GROUP 1662 1663 1664
FACE GROUP 1665 1666 1667
FACE GROUP 1668 1669 1670
FACE GROUP 1671 1672 1673
FACE GROUP 1674 1675 1676
FACE GROUP 1677 1678 1679
FACE GROUP 1680 1681 1682
FACE GROUP 1683 1684 1685
FACE GROUP 1686 1687 1688
FACE GROUP 1689 1690 1691
FACE GROUP 1692 1693 1694
FACE GROUP 1695 1696 1697
FACE GROUP 1698 1699 1700
FACE GROUP 1701 1702 1703
FACE GROUP 1704 1705 1706
FACE GROUP 1707 1708 1709
FACE GROUP 1710 1711 1712
FACE GROUP 1713 1714 1715
FACE GROUP 1716 1717 1718
FACE GROUP 1719 1720 1721
FACE GROUP 1722 1723 1724
FACE GROUP 1725 1726 1727
FACE GROUP 1728 1729 1730
FACE GROUP 1731 1732 1733
FACE GROUP 1734 1735 1736
FACE GROUP 1737 1738 1739
FACE GROUP 1740 1741 1742
FACE GROUP 1743 1744 1745
FACE GROUP 1746 1747 1748
FACE GROUP 1749 1750 1751
FACE GROUP 1752 1753 1754
FACE GROUP 1755 1756 1757
FACE GROUP 1758 1759 1760
FACE GROUP 1761 1762 1763
FACE GROUP 1764 1765 1766
FACE GROUP 1767 1768 1769
FACE GROUP 1770 1771 1772
FACE GROUP 1773 1774 1775
FACE GROUP 1776 1777 1778
FACE GROUP 1779 1780 1781
FACE GROUP 1782 1783 1784
FACE GROUP 1785 1786 1787
FACE GROUP 1788 1789 1790
FACE GROUP 1791 1792 1793
FACE GROUP 1794 1795 1796
FACE GROUP 1797 1798 1799
FACE GROUP 1800 1801 1802
FACE GROUP 1803 1804 1805
FACE GROUP 1806 1807 1808
FACE GROUP 1809 1810 1811
FACE GROUP 1812 1813 1814
FACE GROUP 1815 1816 1817
FACE GROUP 1818 1819 1820
FACE GROUP 1821 1822 1823
FACE GROUP 1824 1825 1826
FACE GROUP 1827 1828 1829
FACE GROUP 1830 1831 1832
FACE GROUP 1833 1834 1835
FACE GROUP 1836 1837 1838
FACE GROUP 1839 1840 1841
FACE GROUP 1842 1843 1844
FACE GROUP 1845 1846 1847
FACE GROUP 1848 1849 1850
FACE GROUP 1851 1852 1853
FACE GROUP 1854 1855 1856
FACE GROUP 1857 1858 1859
FACE GROUP 1860 1861 1862
FACE GROUP 1863 1864 1865
FACE GROUP 1866 1867 1868
FACE GROUP 1869 1870 1871
FACE GROUP 1872 1873 1874
FACE GROUP 1875 1876 1877
FACE GROUP 1878 1879 1880
FACE GROUP 1881 1882 1883
FACE GROUP 1884 1885 1886
FACE GROUP 1887 1888 1889
FACE GROUP 1890 1891 1892
FACE GROUP 1893 1894 1895
FACE GROUP 1896 1897 1898
FACE GROUP 1899 1900 1901
FACE GROUP 1902 1903 1904
FACE GROUP 1905 1906 1907
FACE GROUP 1908 1909 1910
FACE GROUP 1911 1912 1913
FACE GROUP 1914 1915 1916
FACE GROUP 1917 1918 1919
FACE GROUP 1920 1921 1922
FACE GROUP 1923 1924 1925
FACE GROUP 1926 1927 1928
FACE GROUP 1929 1930 1931
FACE GROUP 1932 1933 1934
FACE GROUP 1935 1936 1937
FACE GROUP 1938 1939 1940
FACE GROUP 1941 1942 1943
FACE GROUP 1944 1945 1946
FACE GROUP 1947 1948 1949
FACE GROUP 1950 1951 1952
FACE GROUP 1953 1954 1955
FACE GROUP 1956 1957 1958
FACE GROUP 1959 1960 1961
FACE GROUP 1962 1963 1964
FACE GROUP 1965 1966 1967
FACE GROUP 1968 1969 1970
FACE GROUP 1971 1972 1973
FACE GROUP 1974 1975 1976
FACE GROUP 1977 1978 1979
FACE GROUP 1980 1981 1982
FACE GROUP 1983 1984 1985
FACE GROUP 1986 1987 1988
FACE GROUP 1989 1990 1991
FACE GROUP 1992 1993 1994
FACE GROUP 1995 1996 1997
FACE GROUP 1998 1999 2000
FACE GROUP 2001 2002 2003
FACE GROUP 2004 2005 2006
FACE GROUP 2007 2008 2009
FACE GROUP 2010 2011 2012
FACE GROUP 2013 2014 2015
FACE GROUP 2016 2017 2018
FACE GROUP 2019 2020 2021
FACE GROUP 2022 2023 2024
FACE GROUP 2025 2026 2027
FACE GROUP 2028 2029 2030
FACE GROUP 2031 2032 2033
FACE GROUP 2034 2035 2036
FACE GROUP 2037 2038 2039
FACE GROUP 2040 2041 2042
FACE GROUP 2043 2044 2045
FACE GROUP 2046 2047 2048
FACE GROUP 2049 2050 2051
FACE GROUP 2052 2053 2054
FACE GROUP 2055 2056 2057
FACE GROUP 2058 2059 2060
FACE GROUP 2061 2062 2063
FACE GROUP 2064 2065 2066
FACE GROUP 2067 2068 2069
FACE GROUP 2070 2071 2072
FACE GROUP 2073 2074 2075
FACE GROUP 2076 2077 2078
FACE GROUP 2079 2080 2081
FACE GROUP 2082 2083 2084
FACE GROUP 2085 2086 2087
FACE GROUP 2088 2089 2090
FACE GROUP 2091 2092 2093
FACE GROUP 2094 2095 2096
FACE GROUP 2097 2098 2099
FACE GROUP 2100 2101 2102
FACE GROUP 2103 2104 2105
FACE GROUP 2106 2107 2108
FACE GROUP 2109 2110 2111
FACE GROUP 2112 2113 2114
FACE GROUP 2115 2116 2117
FACE GROUP 2118 2119 2120
FACE GROUP 2121 2122 2123
FACE GROUP 2124 2125 2126
FACE GROUP 2127 2128 2129
FACE GROUP 2130 2131 2132
FACE GROUP 2133 2134 2135
FACE GROUP 2136 2137 2138
FACE GROUP 2139 2140 2141
FACE GROUP 2142 2143 2144
FACE GROUP 2145 2146 2147
FACE GROUP 2148 2149 2150
FACE GROUP 2151 2152 2153
FACE GROUP 2154 2155 2156
FACE GROUP 2157 2158 2159
FACE GROUP 2160 2161 2162
FACE GROUP 2163 2164 2165
FACE GROUP 2166 2167 2168
FACE GROUP 2169 2170 2171
FACE GROUP 2172 2173 2174
FACE GROUP 2175 2176 2177
FACE GROUP 2178 2179 2180
FACE GROUP 2181 2182 2183
FACE GROUP 2184 2185 2186
FACE GROUP 2187 2188 2189
FACE GROUP 2190 2191 2192
FACE GROUP 2193 2194 2195
FACE GROUP 2196 2197 2198
FACE GROUP 2199 2200 2201
FACE GROUP 2202 2203 2204
FACE GROUP 2205 2206 2207
FACE GROUP 2208 2209 2210
FACE GROUP 2211 2212 2213
FACE GROUP 2214 2215 2216
FACE GROUP 2217 2218 2219
FACE GROUP 2220 2221 2222
FACE GROUP 2223 2224 2225
FACE GROUP 2226 2227 2228
FACE GROUP 2229 2230 2231
FACE GROUP 2232 2233 2234
FACE GROUP 2235 2236 2237
FACE GROUP 2238 2239 2240
FACE GROUP 2241 2242 2243
FACE GROUP 2244 2245 2246
FACE GROUP 2247 2248 2249
FACE GROUP 2250 2251 2252
FACE GROUP 225
```

## VII. SUMMARY AND CONCLUSIONS

### 7.1 Summary

The objective of this research was to investigate techniques for the computer understanding of pictures of scenes. Three important principles guided the research: (a) No complete preknowledge of the scenes' bodies should be presumed since that requires a limited repertoire of bodies. Instead, the number of bodies should be unrestricted and merely some general properties of the bodies should be prespecified. (b) These properties should be flexible enough to permit inclusion of all bodies likely to be encountered. (c) Since the input is to be gotten from photographs, the presence of data imperfections such as missing information, wrong information, and geometric inaccuracies must be taken into consideration. The difficulty of the described objective is highlighted by the fact that previous research in this area always presupposed the placing of severe restrictions on the problem environment. Where imperfect data was dealt with, only a small, fixed repertoire of polyhedra was allowed (6). Where the fixed repertoire was replaced by general properties of a family of bodies (though less general than those assumed in this thesis), perfect input data was required (3, 18).

To achieve the objective set here, set of multiple pictures of the scene was assumed to serve as the input data. New grammar rules needed for the analysis of these pictures were formalized. A procedure for establishing matches between features in the different pictures, and for verifying doubtful features, were devised, as well as procedures for eliminating wrong data and for recovering missing data. Finally, a



procedure for assembling the analyzed data into sets, each describing a single body of the scene, was constructed, including the determination of the bodies' faces and their nature.

A computer program based on this approach was written. The program was able, given real input data, to "understand" the photographed scenes, and to yield plausible descriptions. The program is cautious in the decision making process, at the expense of losing some of the given features. Thus, although the description obtained may be incomplete, no errors are expected. A common and crucial difficulty encountered in the preprocessing stage is telling a T junction from a flat Y or W junction. This can be handled by the program, if desired, by adding the flat Y and W junctions to the junctions that need verification. No restrictions need be placed on the intersection of two curved faces; however, if the intersection is of a degree higher than that of a conic, the determination of an A junction may be problematic.

The main weakness of the test of the program is that the program was not linked to an existing preprocessor. It thus was not subjected to a completed "reality" test. However, if and when such a link-up with a preprocessor is achieved, the power of the main program will grow in the following sense: In the stage of data recovery we made a final judgment as to whether or not a line should be extended to a junction on the basis of the fitted equation and a set of severe restrictions. Once the program is connected to a preprocessor, this can be replaced with a request from the program to the preprocessor to take a closer look in the particular area of interest. The data recovery decision can then be made on the basis of new more precisely determined data.

During the course of the development of the matching procedure, it became apparent that a set of three pictures is a good choice but not optimal. If we use a bigger set of pictures (covering approximately the same visible range), we are likely to need less context support and are almost certain to obtain superior results.

## 7.2 Specific Contributions

The following are considered to be the specific contributions presented in this dissertation.

1. New grammar rules, crucial for picture analysis, for a family of curved bodies were formulated. These rules, when also applied to polyhedra, complement the set of rules already known and strengthen our ability for machine analysis of pictures containing polyhedral-type bodies.
2. Procedures were developed for matching features in a set of pictures taken from different vantage points, taking into consideration the possible presence of serious errors in feature extraction.
3. A technique was developed for the verification of doubtful features extracted from the pictures, and for the elimination of some possible preprocessor errors.
4. Data correction and recovering techniques were developed having the ability of knowledgeably instructing the preprocessor for further data extraction efforts.



5. A procedure was formulated for assembling the analyzed data from all pictures into sets, each representing a body, and segmenting into subsets, each representing a face. In addition a procedure was given for determining the nature of each face (i.e., whether planar or curved and what type of curve if curved).
6. A program was written and tested for describing scenes of curved bodies on the basis of imperfect data extracted from multiple pictures. Except for information about some general properties, the program has no a priori knowledge of the bodies. The program consists of about 3,500 PL/I statements and was tested against data extracted from photographs of real scenes.

### 7.3 Directions for Future Research

In our search to provide computers with some kind of 3D perception, there is a strong tendency to imitate the way a human being's perception abilities function. There are two difficulties with that approach. Firstly we do not know how a human's perception really works. All models for human perception presented thus far at best are able to explain only a small part of this ability. Secondly it is questionable as to whether imitating human perception is the best approach for developing machine perception. It is the writer's feeling that a breakthrough in machine perception, which departs from the traditional approach, is needed and due to come.

Meanwhile, following the present approach it appears that considerable improvements in performance can be achieved if the preprocessor is incorporated more intimately into the overall process. It should not stand by itself but rather be an integral part of the whole picture-understanding process. For example, in the process of determining an intersection of lines to locate a junction (a painful task for the preprocessor) other pictures should be consulted, utilizing the geometric relations described in the thesis. The process of extracting features from the pictures should not be serial but rather parallel, and thoroughly exploiting the matching tools described. Also, as already mentioned in Section 7.1, the possibility should be considered of using pictures from more than three vantage points to provide the data for the scene analysis. Finally, thought should be given to broaden and unify the scene description terms. The two present modes of description - faces and their intersections and combinations of simple volumes - should be used together, in addition to other future terms, in a formal scene description language.



## REFERENCES

1. G. J. Agin, "Representation and Description of Curved Objects," doctoral dissertation, Computer Science Department, Stanford University, Stanford, CA., October 1972.
2. A. Albano, "Representation of Digitized Contours in Terms of Conic Arcs and Straight-Line Segments," Computer Graphics and Image Processing, 3, (1), 1974, 23-33. (AD 766 315).\*
3. R. T. Chien, Y. H. Chang, "Recognition of Curved Objects and Object Assembly," Proc. of the Second International Joint Conference on Pattern Recognition, IEEE publication No. 74 CH0885-4C, Copenhagen, pp. 496-510, August 1974.
4. M. B. Clowes, "On Seeing Things," Artificial Intelligence, 2, (1), 1970, 79-116.
5. R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, John Wiley Interscience, New York, 1973.
6. G. Falk, "Computer Interpretation of Imperfect Line Data as a Three-Dimensional Scene," doctoral dissertation, Computer Science Department, Stanford University, Stanford, CA, August 1970. (AD 715 665).
7. A. Guzman, "Computer Recognition of Three-Dimensional Objects in a Visual Scene," doctoral dissertation, M.I.T., Cambridge, Mass., December 1968. (AD 692 200).
8. D. A. Huffman, "Impossible Objects as Nonsense Sentences," Machine Intelligence, 6, ed. B. Meltzer and B. Michie, American Elsevier, 1971, 295-323.
9. A. Rabinowitz, "Reconstruction of Polyhedra from Sets of this Perspective Projections," doctoral dissertation, Dept. of Electrical Engineering, New York University, April 1971. (AD 732 300).
10. U. Ramer, "Computer Edge Extraction from Photographs of Curved Objects," doctoral dissertation, School of Engineering, New York University, N.Y., November 1973. (AD 778 456)
11. L. G. Roberts, "Machine Perception of Three-Dimensional Solids," Optical and Electrooptical Information Processing, ed. J. T. Tippett et. al., M.I.T. Press, Cambridge, Mass., pp. 159-197, 1965.

---

\* Items marked with "AD" numbers can be obtained for a nominal charge from NTIS, U. S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

12. R. Shapira, "A Technique for the Reconstruction of a Straight-Edge, Wire-Frame Object from Two or More Central Projections," Computer Graphics and Image Processing, 3, (4), 1974, 318-326. (AD 751 282).
13. R. Shapira and H. Freeman, "A Cyclic-Order Property of Bodies with Three-Face Vertices," Tech. Rept. CRL-42, Div. Applied Science, New York University, August 1975.
14. R. Shapira, "Computer Reconstruction of Quadric-Surfaced Objects from Two or More Visible-Line Projections," Tech. Rpt. CRL-44, Div. Applied Science, New York University, August 1975. (AD-A020 628).
15. Y. Shirai, S. Tsuji, "Extraction of the Line Drawings of Three-Dimensional Objects by Sequential Illumination from Several Directions," Proc. Second International Joint Conf. on Artif. Intelligence, British Computer Society, London, 71-79, 1971.
16. L. B. Smith, "The Use of Man-Machine Interaction in Data-Fitting Problems," doctoral dissertation, Stanford University, Stanford, CA, March 1969.
17. I. Sobel, "Camera Models and Machine Perception," Doctoral Thesis, Artificial Intelligence Memo No. 121, Stanford University, Stanford CA, May 1970. (AD 708 084).
18. D. L. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," doctoral dissertation, Artif. Intelligence Laboratory, M.I.T., Cambridge, Mass., November 1972. (AD 754 080).
19. S. A. Underwood, C. L. Coates Jr., "Visual Learning from Multiple Views," IEEE Transaction on Computers, C-24, No. 6, June 1975.
20. P. Woon, "A Computer Procedure for Generating Visible Line Drawings for Solids Bounded by Quadric Surfaces," doctoral dissertation, Dept. of Electrical Engineering, New York University, December 1970. (AD 724 744).
21. P. Woon, H. Freeman, "A Procedure for Generating Visible Line Projections of Solids Bounded by Quadric Surfaces," Information Processing 71, North-Holland Pub. Co., Amsterdam, 1120-1125, August 1972.



INDEX OF DEFINITIONS

<u>Definition</u>	<u>Page</u>
Body	5
Boundary	5
Cyclic Order	11
Depth Index	11
Edge	5
Empty Line	49
Face	5
Face Group	60
Junction	6
Limb	5
Line	6
Line Assembly (LA)	13
Match Line	24
Match Triple	28
Natural Extension	41
Object	6
Projection	5
Range Limits	57
Region	6
Scene	5
Surface	5
Synthetic Junction	51
Valid Junction	23
Vertex	5
Virtual Vertex	5

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER <b>AFOSR-TR-77-0051</b>		2. GOVT ACCESSION NO.	
3. TITLE (and Subtitle) <b>COMPUTER RECONSTRUCTION OF BODIES BOUNDED BY QUADRIC SURFACES FROM A SET OF IMPERFECT PROJECTIONS.</b>		4. RECIPIENT'S CATALOG NUMBER	
5. AUTHOR <b>Ruth/Shapira</b>		6. TYPE OF REPORT & PERIOD COVERED <b>Interim rept</b>	
7. PERFORMING ORGANIZATION NAME AND ADDRESS <b>Electrical and Systems Engineering Dept Rensselaer Polytechnic Institute Troy, NY 12181</b>		8. PERFORMING ORG. REPORT NUMBER <b>19 CRL-748</b>	
9. CONTROLLING OFFICE NAME AND ADDRESS <b>Air Force Office of Scientific Research/NM Bolling AFB, DC 20332</b>		10. CONTRACT OR GRANT NUMBER(s) <b>15 VAF-AFOSR-2937-76, VAF-AFOSR-2755-75</b>	
11. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS <b>61102F 17 2304 A2</b>	
13. DISTRIBUTION STATEMENT (of this Report) <b>Approved for public release; distribution unlimited.</b>		14. REPORT DATE <b>16 Sep 76</b>	
15. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		15. NUMBER OF PAGES <b>116</b>	
16. SUPPLEMENTARY NOTES		16. SECURITY CLASS. (of this report) <b>UNCLASSIFIED</b>	
17. KEY WORDS (Continue on reverse side if necessary and identify by block number) <b>scene analysis impossible objects artificial intelligence picture grammars image processing 3D object reconstruction</b>		17. DECLASSIFICATION/DOWNGRADING SCHEDULE	
18. ABSTRACT (Continue on reverse side if necessary and identify by block number) <b>This thesis describes a computer program for constructing a description of solid bodies from a set of n pictures of the bodies. The bodies are assumed to be bounded by faces which are quadric or planar, and they are restricted to have all their vertices formed by exactly three faces. The pictures are taken from different vantage points, with the restriction that a slight shift in vantage point will not alter the topology of the picture. It is assumed that the program re- ceives outline information from a preprocessor which has extracted this</b>			



information from the pictures. The outline information (set of line structures) may be imperfect in that some junctions may be erroneously reported and some lines may be missing. However, all lines due to shadows are assumed to have been eliminated by the preprocessor.

The thesis includes a technique for establishing the validity of the junctions presented by the preprocessor as well as for matching corresponding features in the line structures derived from the different pictures. New grammar rules for line-drawing projections of curved and planar solid bodies are developed. These are useful in parsing the line drawings. They have also led to the definition of a new family of impossible objects. The program works simultaneously with all the available line structures. The parsing of every line structure is supported dynamically by the results gotten thus far from the parsing of the other line structures. Through the parsing of the line structures the use of picture comparison and the application of the grammar rules, many of the preprocessor errors are detected and partly corrected. The program also can provide feedback to the preprocessor in the form of suggestion as to where to look again for lines in the pictures.

The program utilizes the extracted line structures corresponding to the different bodies in all the pictures to determine the set of faces (insofar as possible) for every body. Every face is defined by an ordered set of  $n$ -tuples. The  $n$ -tuples are the matched lines and junctions in the  $n$  different pictures. The three-dimensional coordinates of the vertices and the equations of the faces can then be determined from these  $n$ -tuples. The program was written in PL/I and has been tested on several scenes.

UNCLASSIFIED